

SET – gruppe 4 prosjektdokumentasjon

Magnus, Anders, Sigmund, Mathias, Andreas

Innholdsfortegnelse

Innhold

Innholdsfortegnelse	1
Introduksjon	6
Teknologi	6
Målgruppe	6
Eksterne linker	6
Brukertyper	6
Generell Bruker	6
Beskrivelse	6
Persona	6
Use case diagram	7
Use cases	7
User Story	8
Arrangør	8
Beskrivelse	8
Persona – Aktuell person	9
Persona - Aktuell organisasjon	9
Use case diagram	9
Use cases	10
User story	10
Gruppeleder	10
Beskrivelse	10
Persona (gjelder gruppe- og klubbleder)	10
Use case diagram	11
Use cases	11
User story	11
Klubbleder	12
Beskrivelse	12
Use case diagram	12
Use cases	12

Gruppemedlem/klubbmedlem	12
Beskrivelse	12
Persona	13
Use case diagram.....	13
Use cases	13
User story	13
Serviceadministrator	14
Beskrivelse	14
Persona	14
Use case diagram.....	14
Use cases	14
User story	15
Databaseadministrator.....	15
Beskrivelse	15
Persona	15
<i>Use case diagram</i>	15
Use cases	15
User story	16
Funksjonelle krav.....	16
Brukergrensesnitt.....	16
Bruker.Brukergrensesnitt.Skal kunne se	16
Klubbleder.Brukergrensesnitt.Skal kunne se	16
Gruppeleder.Brukergrensesnitt.Skal kunne se.....	17
Gruppemedlem/Klubbmedlem.Brukergrensesnitt.Skal kunne se	17
Registrering	17
Systemadministrator.Registerer.Skal ikke kunne.....	18
Innlogging	18
Sortering.....	18
Meldinger	18
Alle brukere. Sender melding til meldingsgruppe.Skal kunne	18
Arrangør.Sender melding.Skal kunne.....	19
Bruker.Sender melding.Skal kunne	19
Arrangør.Sender medling.Skal kunne.....	19
Klubbleder/Gruppeleder.Sender melding.Skal kunne	19
Serviceadministrator.Sender melding.Skal kunne	19
Brukerspesifikke krav	19

Bruker	19
Bruker.Påmelde arrangement.Skal kunne	19
Bruker.Påmelde arrangement.Skal ikke kunne melde på arrangementer.....	20
Bruker.Avmelde arrangement.Skal kunne melde av arrangementer	20
Bruker.Opprette gruppe.Skal kunne	20
Bruker.Behandle invitasjon.Skal kunne.....	20
Klubbleder	20
Klubbleder.Administrerer klubb.Skal kunne	20
Klubbleder.Administrerer klubb.Skal ikke kunne	20
Klubbleder.Påmelder klubb.Skal kunne	20
Klubbleder.Påmelder klubb.Skal ikke kunne	21
Klubbleder.Melder klubb av arrangement.Skal kunne	21
Klubbleder.Melder klubb av arrangement.Skal ikke kunne	21
Gruppeleder	21
Gruppeleder.Søker som klubbleder.Skal kunne.....	21
Klubbmedlem/gruppemedlem.....	21
Medlem.Administrere.Skal kunne.....	21
Medlem.Administrere medlemskap.Skal ikke kunne.....	21
Arrangør	21
Arrangør.Opprette arrangement.Skal kunne	21
Arrangør.Opprette arrangement.Må velge.....	22
Arrangør.Opprette arrangement.Skal ikke kunne.....	22
Arrangør.Administrere arrangement.Skal kunne.....	22
Arrangør.Administrere arrangement.Skal ikke kunne	22
Administratorer	22
Serviceadministrator.Skal kunne	23
Serviceadministrator.Skal ikke kunne	23
Databaseadministrator.Skal kunne	23
Systemkrav	23
System.Innlogging.Skal	23
System.Arrangør.Skal	23
System.Bruker.Skal	24
System.Bruker.Skal ikke	24
System.Gruppelider.Skal	24
System.Klubbleder.Skal	25
System.Klubbmedlem/Gruppemedlem.Skal	25

System.Serviceadministrator.Skal	25
System.Gruppe/Klubb opprettelse.Skal	25
Ikke funksjonelle krav	25
System	25
Databaserelaterte krav	25
Ytelsesrelaterte krav	26
Sikkerhetsrelaterte krav	26
Gjenopprettelserelaterte krav	26
Safety relaterte krav	26
Redundansrelaterte krav	26
Tilgjengelighetskrav	26
Lovpålagt	26
Ikke lovpålagt	27
Estimering og prioritering av funksjonalitet	27
Brukerfunksjoner	27
Klubb/Gruppe -lederfunksjoner	28
Klubb/Gruppe – medlemsfunksjoner	28
Arrangørfunksjoner	28
Serviceadministratorfunksjoner	29
Systemfunksjoner	29
Tredjepartsintegreringer	30
Betaling	30
Klarna	30
Innlogging	31
Facebook og Google	31
Andre tredjepartsintegrasjoner	32
Statens organisasjonsregister	32
Google maps	34
ID-brikke	34
Arkitektur og sammenhenger i systemet	34
Introduksjon	34
Beskrivelse av partene	35
Front end	35
Back end – Distribusjon av oppgaver	35
Back end – Logikk og funksjonalitet	35
Database – Lagring av data	37

Prototype.....	38
Formålet med prototypen.....	38
Valgt teknologi.....	38
Valgte brukertyper	38
Valgte funksjonelle krav	39
Valgte ikke funksjonelle krav.....	39
Data Definisjon	40
Beskrivelse	40
Bruker	40
Innlogging	40
Arrangement	40
Grupper	41
Idrettskategorier.....	41
Styrker	41
Svakheter.....	41
Sekvensdiagram – Prototype.....	42
Tester.....	43
UseCase	43
Unit Testing	44
Grupper:	45
Brukergrensesnitt	45
Betalingsintegrasjon stubb.....	47
<i>Klarna</i>	47
Brukermanual for Prototype	48
Login	48
Login som bruker.....	48
Login som arrangør.....	50
Login som admin	50
Hvordan kjøre prototypen.....	51
Ordforklaringer.....	51

Introduksjon

Systemet tar for seg oppgaven med å løse et ønske stilt av ulike idrettsforbund knyttet mot arrangementer for fysisk aktivitet og forskjellig type idretter. Det handler om mindre idrettslag og -klubber som oftest ikke har kapasitet til å håndtere infrastrukturen rundt arrangementene sine selv. De ønsker derfor et felles system som løser dette.

Systemet skal håndtere alt fra da *brukeren* oppdager arrangementet, til etter arrangementet er avsluttet. Dette inkluderer følgende aspekter: *brukerhåndtering*, registrering/opprettelse av arrangementer samt påmelding til disse. Videre skal systemet håndtere registrering av grupper og verifisering av klubber, samt enkel kommunikasjon mellom *klubb/gruppeledere* og medlemmer. Systemet skal i tillegg lagre og fremvise resultater fra avsluttede arrangementer, og ha muligheten for delinger av arrangementer gjennom sosiale medier.

Systemet skal dermed fungere som ett sentralt system for lokale og nasjonale idrettsforbund. Derunder oppgaver som påmelding og spredning av informasjon som angår arrangementer. Dette gjelder også innad i idrettsklubber og deres arrangementer. Videre skal *brukere* kunne ha muligheten til å utforske samt registrere seg for nye arrangementer. Derav inkluderer dette muligheten til å søke og sortere etter gitt navn eller gitt kategori for arrangementer.

Teknologi

Løsningen vil primært være en webapplikasjon, noe som øker tilgjengeligheten og utbredelsen av applikasjonen. Grensesnittet blir da et webgrensesnitt. Videre vil systemets funksjonalitet og hovedlogikk ligge på serversiden. I tillegg til dette vil data ligge i en database håndtert av et DBMS. Innhenting av resultater foretas primært gjennom selve arrangementet, gitt ved utdelte ID brikker.

Målgruppe

Denne dokumentasjonen er ment lest av følgende grupper: Eksternt utviklingsselskap, produkteier og oppdragsgiver.

Eksterne linker

WCAG 2.0: <https://uu.difi.no/krav-og-regelverk/wcag-20-standarden>

GDPR: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2016:119:FULL>

Angrerettloven kapittel 5: https://lovdata.no/dokument/NL/lov/2014-06-20-27#KAPITTEL_5

Brukertyper

Generell Bruker

Beskrivelse

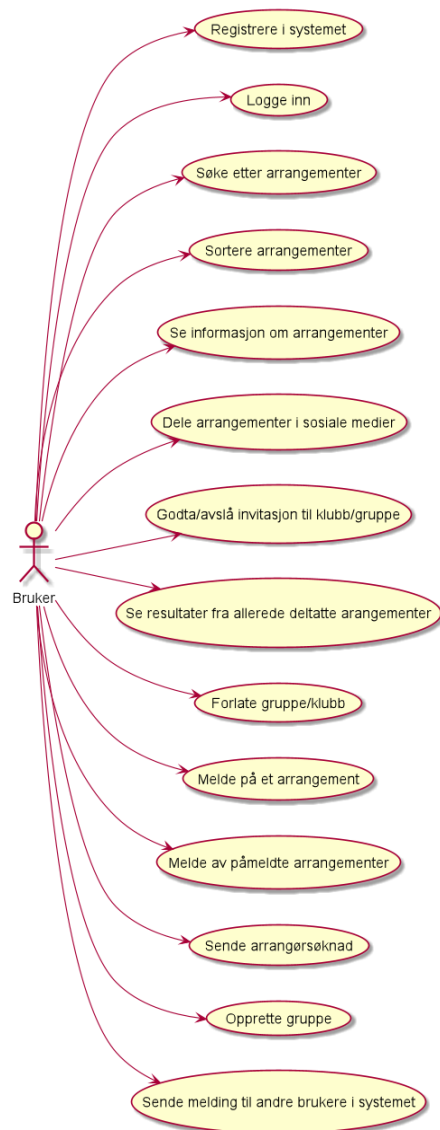
En tilfeldig registrert person som ønsker å benytte systemet. Derunder basisfunksjoner som leting og påmelding til arrangementer, men involverer ikke opprettelse eller administrering av systemet. Kompetansen til *brukerne* er alle fra de med liten datakyndighet til de veldig kyndige.

Persona

Navn	Lille Ole
------	-----------

Alder	12 år
Skole	Barneskole, 7. trinn
Interesser	Sykkel, Ski og Gaming
Bruksområde i systemet	Benytte systemet til å delta på idrettsarrangementer både alene og i grupper.
Brukergruppe	<i>Bruker</i>

Use case diagram



Use cases

1. Som en *bruker* ønsker jeg å kunne registrere meg i systemet og logge inn slik at jeg kan benytte meg av systemets funksjoner.

2. Som en *bruker* ønsker jeg å kunne søke opp og sortere arrangementer etter arrangement navn, navn på *arrangør*, lokasjon, type idrett og start- og sluttdato, slik at jeg kan melde meg på det arrangementet som er mest relevante for meg.
3. Som en *bruker* ønsker jeg å kunne godta/avslå invitasjon til en gruppe utfra om jeg ønsker å være med eller ikke.
4. Som en *bruker* ønsker jeg å se resultater fra mine allerede deltatte arrangementer.
5. Som en *bruker* ønsker jeg å kunne dele arrangementer på sosiale medier slik at jeg kan inkludere vennene mine, holde de oppdatert på hva jeg skal delta på og kanskje også gjøre de interessert i samme arrangement.
6. Som en *bruker* ønsker jeg å bli varslet om endringer som skjer i en gruppe/klubb via mail slik at jeg holder meg oppdatert på endringer.
7. Som en allerede registrert *bruker* ønsker jeg videre å kunne registrere meg som en *arrangør* slik at jeg slipper å lage ny *bruker* for å bli *arrangør* om jeg i senere tid også vil arrangere arrangementer.
8. Som en *bruker* ønsker jeg å kunne melde meg av et arrangement slik at jeg slipper å møte opp samtidig som *arrangør*er unngår at påmeldte ikke møter opp.
9. Som en *bruker* ønsker jeg å kunne sende meldinger til andre i systemet slik at kommunikasjon rundt deltagelse av arrangementer og lignende blir lettere.
10. Som en *bruker* ønsker jeg å kunne opprette grupper slik at jeg kan ha muligheten til å delta med andre.

User Story

Lille Ole er 12 år og interessert i sykkel og ski. Han ønsker å benytte systemet til å delta på et idrettsarrangement med noen venner. Han velger dermed å registrere seg i systemet med nødvending registreringsinformasjon som navn, epost, passord og alder, slik at han kan søke etter arrangement. Han benytter seg av søke- og sorteringsfunksjonene for å finne arrangementer som passer best for han. Når han har funnet et passende arrangement vil han dele det på sosiale medier slik at han kan finne noen å delta med. Han oppretter en gruppe og legger til medlemmer i gruppen utfra de som viste interesse for det arrangementet han delte på sosiale medier. For å sørge for at alle møter opp, sender han ut en notifikasjon via gruppen til alle som er påmeldt arrangementet 24-timer før. Når arrangementet er over, sjekker alle resultatene sine (tid brukt) og sammenligner det med de andre på gruppen.

Scenariet over beskriver en mulighet for hvordan systemet kan bli brukt som en *bruker*. Ut av dette kan følgende funksjoner vises.

- Registrering som *bruker*
- Sorterings- og søkefunksjoner for arrangementer
- Dele arrangementer på sosiale medier
- Oppretting av en gruppe
- Legge til medlemmer i gruppe
- Automatisk notifikasjon til gruppemedlemmer 24-timer før arrangementstart.
- Visning av resultater, både for enkeltpersoner og gruppen.

Arrangør

Beskrivelse

En person/organisasjon som står ansvarlig for administrering og opprettelse av ett eller flere arrangementer. For at en *bruker* skal få status som *arrangør*, kreves det utfylling av søknad som

sendes videre inn til produkteier sin bedrift. Følgelig er det nevnt bedrift sine retningslinjer som styrer kravet som stilles til *arrangør*.

Persona – Aktuell person

Navn	Martin
Alder	37 år
Jobb	Barnehageansatt
Interesser	Jogging, strikking og skogsturer.
Bruksområde i systemet	Benytte systemet for å håndtere infrastrukturen til arrangementer.
Brukergruppe	<i>Arrangør</i>

Persona - Aktuell organisasjon

Navn	Tore's idrettsarrangementer
Bransje	Organiserer renn og løps -idretter.
Bruksområde i systemet	Benytte systemet for å håndtere infrastrukturen til arrangementer.
Brukergruppe	<i>Arrangør</i>

Use case diagram



Use cases

1. Som *arrangør* ønsker jeg å få informasjon om arrangementene mine, slik som resultater fra *brukerne* og antall påmeldt slik at jeg får god oversikt.
2. Som *arrangør* ønsker jeg å kunne opprette arrangementer med informasjon som navn, idrettstype/kategori, maks antall plasser, startdato, sluttdato, startklokkeslett, slutt klokkeslett og om det er gruppe/klubb arrangement eller for enkeltpersoner, slik at jeg får en god beskrivelse passende til arrangementet.
3. Som en *arrangør* ønsker jeg å kunne velge min ønsket betalingstjeneste slik at jeg kan få inn betaling på riktig måte.
4. Som *arrangør* ønsker jeg å kunne redigere arrangementer slik at jeg kan rette opp feil.
5. Som *arrangør* ønsker jeg å få oversikt over resultatene fra alle *brukerne* som har deltatt i arrangementet mitt.
6. Som en *arrangør* ønsker jeg muligheten til på sende meldinger til andre i systemet slik at kommunikasjon rundt administrering av arrangementer blir lettere.

User story

Martin er en 37 år gammel barnehageansatt som vil benytte systemet til opprettelse av et arrangement på vegne av jobben sin. Som en allerede registrert bruker, velger han videre å sende inn en arrangørsøknad. Ved godkjenning logger han inn og velger «Nytt arrangement» for deretter å legge inn nødvendig informasjon for registrering av det nye arrangementet. Etter at arrangementet er ferdig ønsker han å gi arrangementeierskapet/administratorrollen til en annen ansatt. Den nye eieren oppdager en feil i beskrivelsen og trykker derfor «Rediger» for deretter å rette opp i feilen. Han ser så gjennom informasjon om alle deltagerne for å sjekke at alt er klart.

Scenariet over beskriver en mulighet for hvordan systemet kan bli brukt som en *arrangør*, og tester disse funksjonene for systemet:

- Innsendelse og godkjenning av en arrangørsøknad
- Opprettelse av nytt arrangement
- Endring i arrangement eierskap
- Redigering av allerede opprettet arrangement
- Se oversikt over *brukere* påmeldt et arrangement

Gruppeleder

Beskrivelse

Bruker som oppretter en gruppe. Denne *brukeren* innehar ansvaret for administrering av gruppen og dets medlemmer, samt påmelding til arrangementer og utsending av gruppenotifikasjoner.

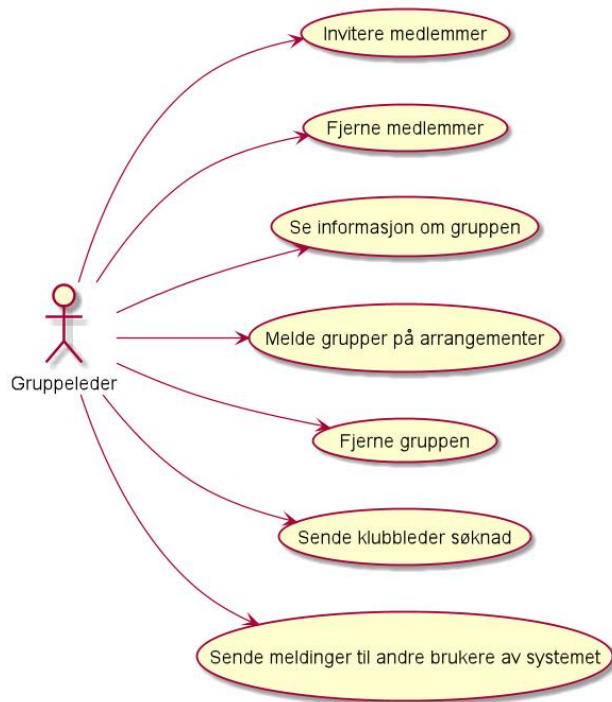
En Gruppeleder vil fremdeles ha de samme rettighetene som *Bruker*.

Persona (gjelder gruppe- og klubbleder)

Navn	Lars Leder
Alder	56 år
Yrke	Pensjonist
Utdanning	Selvstendig næringsdrivende
Interesser	Fotball, liker å lede

Bruksområde i systemet	Administrere gruppen han er leder for
Brukergruppe	Gruppeleder/klubbleder

Use case diagram



Use cases

1. Som *gruppeleder* ønsker jeg å registrere meg som en *klubbleder*/sende inn *klubbsøknad* dersom gruppen er en faktisk klubb.
2. Som *gruppeleder* ønsker jeg å kunne legge slik at de ikke trenger å melde seg på selv.
3. Som *gruppeleder* ønsker jeg å kunne fjerne *medlemmer* dersom jeg ikke ønsker å ha de med i gruppen.
4. Som *gruppeleder* ønsker jeg å melde gruppen på et arrangement slik at jeg alene kan administrere hva vi skal delta i uten at alle *enkeltmedlemmer* trenger å melde seg på.
5. Som *gruppeleder* ønsker jeg å se informasjon om gruppen og *medlemmene* slik at jeg har oversikt over hvem som er *medlem* og hva gruppen er meldt på samt tidligere historikk.
6. Som en *gruppeleder* ønsker jeg muligheten til å sende meldinger til andre medlemmer slik at jeg lettere kan kommunisere med gruppen.

User story

Lars Leder er 56 år og liker å trene og styre idrettslag og klubber på fritiden. Han ønsker å benytte seg av systemet for å administrere påmelding av et av idrettslagene til arrangementer. Han oppretter en bruker i systemet og deretter en gruppe. Ettersom gruppen er en offisiell idrettsklubb, sender han inn en klubbsøknad og får den godkjent. Han legger deretter til medlemmer i klubben. For de som ikke allerede er brukere i systemet, oppgir han en epost-adresse som vil motta en registreringslink. Han sjekker oversikten over klubben og melder de deretter på idrettsarrangementet.

Scenariet over beskriver en mulighet for hvordan systemet kan bli brukt som en *gruppe-/klubbleder*, og tester disse funksjonene for systemet:

- Registrering som *bruker*
- Opprettelse av gruppe
- Innsendelse og godkjenning av klubbsøknad
- Legge til *medlemmer* i gruppe/klubb
- Registreringslink via epost
- Se klubb-/gruppeoversikt
- Påmelding av klubb til arrangement

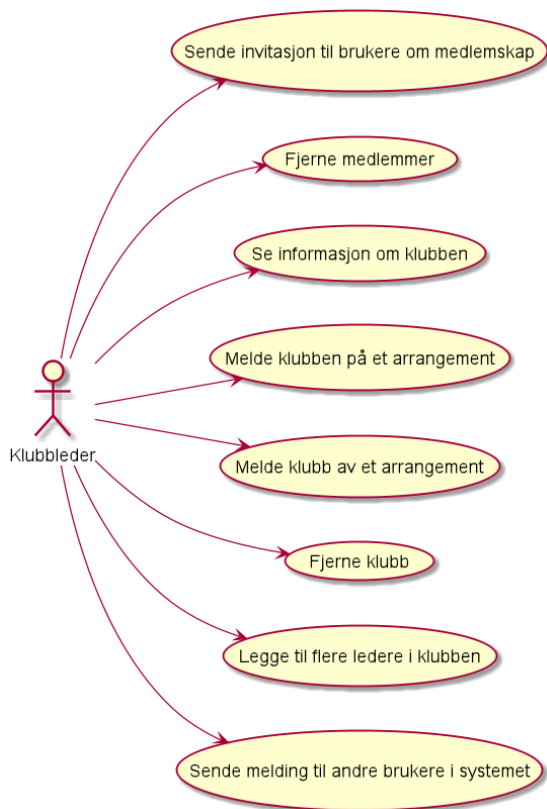
Klubbleder

Beskrivelse

Henviser til beskrivelsen av *Gruppeleder*. Dette er en *gruppeleder* tilhørende en klubb som har blitt autorisert som en faktisk klubb. En autorisert klubb vil typisk være et større idrettslag istedenfor en gruppe.

En Klubbleder vil fremdeles ha de samme rettighetene som *Bruker*.

Use case diagram



Use cases

1. Som *klubbleder* ønsker jeg å kunne legge til flere ledere for klubben slik at jeg alene ikke trenger å administrere den.

[Use case 2, 3, 4 og 5 fra *Gruppeleder*](#) gjelder også for *Klubbleder*.

Gruppemedlem/klubbmedlem

Beskrivelse

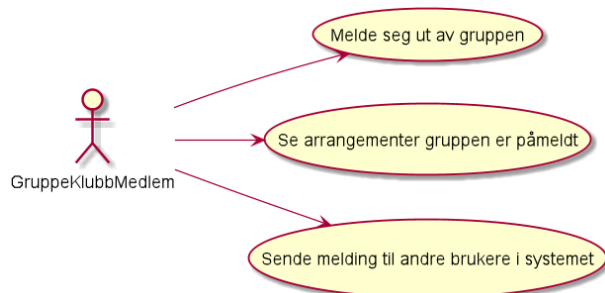
En *Bruker* som er *medlem* av en gruppe eller klubb.

En Klubbleder vil fremdeles ha de samme rettighetene som *Bruker*.

Persona

Navn	Kåre <i>Klubbmedlem</i>
Alder	29 år
Yrke	NAV
Utdanning	Idrettslinje VGS
Interesser	Løpe, sykle, ski, alt som heter idrett
Bruksområde i systemet	Benytte klubb delen av systemet
Brukergruppe	<i>Gruppemedlem/klubbmedlem</i>

Use case diagram



Use cases

1. Som et *medlem* av en gruppe/klubb ønsker jeg muligheten til å se hvilke arrangementer gruppen/klubben er påmeldt slik at jeg får oversikt over fremtidige arrangementer.
2. Som et *medlem* av en gruppe/klubb ønsker jeg muligheten til å melde meg ut av gruppen om jeg ikke ønsker å være medlem lenger.
3. Som et *medlem* av en gruppe/klubb ønsker jeg muligheten til å sende meldinger til andre medlemmer i gruppen/klubben slik at vi lettere kan kommunisere.
4. Som en *gruppe/klubbmedlem* ønsker jeg muligheten til å sende meldinger til andre brukere slik at jeg lettere kan kommunisere med gruppen.

User story

Kåre Klubbmedlem er 29 år og veldig interessert i idrettsaktiviteter. Han er derfor med i en rekke idrettsklubber som benytter seg av dette systemet. Han går gjennom hver klubb og sjekker oversikten over hvilke arrangementer de er meldt på slik at han vet hva som skjer fremover. Noen av arrangementene han ikke har tid til velger han å melde seg av, selv om han fortsatt vil være medlem i klubben. Ettersom han er medlem av såpass mange grupper/klubber har han bestemt seg for å melde seg ut av noen grupper/klubber.

Scenariet over beskriver en mulighet for hvordan systemet kan bli brukt som en *gruppe-/klubbmedlem*, og tester disse funksjonene for systemet:

- Se oversikt over påmeldte arrangementer for grupper/klubber
- Melde seg av arrangement individuelt, selv om resten av klubben/gruppen deltar
- Melde seg ut av gruppe/klubb

Serviceadministrator

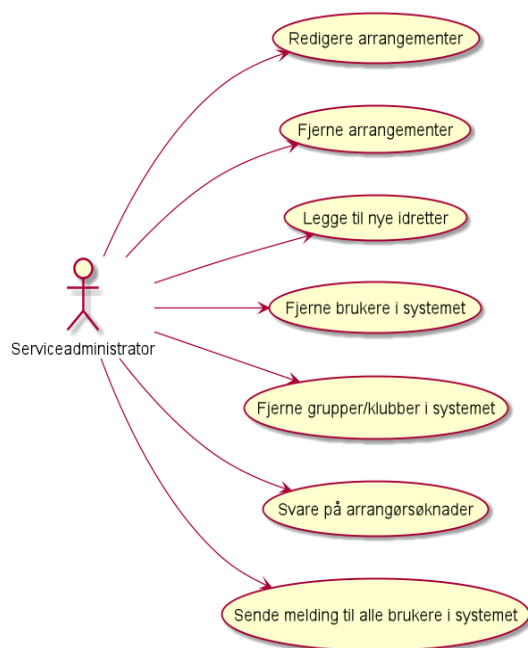
Beskrivelse

En *administrator* som håndterer service aspektet opp mot *brukere* og arrangementer. Personen har dermed fullmakt til å utføre endringer relatert til arrangementer samt registreringer av *arrangører*. Dette vil være en ansatt som utfører pliktige oppgaver for produktinnehaver av systemet.

Persona

Navn	Bernt Arne
Alder	34 år
Yrke	Jobber i firma som er vert og drifter IT - systemer
Utdanning	Fagbrev, IKT-servicefag.
Interesser	Gaming, Svømming og jakt
Bruksområde i systemet	Administrere arrangementer og brukergrupper.
Brukergruppe	<i>Serviceadministrator</i>

Use case diagram



Use cases

1. Som *serviceadministrator* ønsker jeg å kunne fjerne og redigere arrangementer samt fjerne arrangement-statusen på *brukere* slik at jeg kan sikre meg at alle arrangementer og *arrangører* holder seg seriøse.
2. Som *serviceadministrator* ønsker jeg å kunne legge til idrettskategorier slik at arrangementer kan bli bedre presisert dersom ikke idrettstypen allerede går innunder en allerede definert kategori.
3. Som *serviceadministrator* ønsker jeg å kunne redigere arrangementer. Dette vil kunne føre til redusering av mulige XSS, samt endre støtende beskrivelser.
4. Som *serviceadministrator* ønsker jeg å svare på arrangørsøknader slik at systemet får flere *arrangører* samtidig som jeg kan filtrere bort de som ikke passer til denne rollen.

5. Som *serviceadministrator* ønsker jeg muligheten til å sende meldinger til brukere i systemet slik at jeg lett kan informere de om endringer.

User story

Bernt Arne er en 34 år gammel *serviceadministrator* for systemet som har oppgaven å administrere arrangementer og brukergrupper. Han går gjennom alle nylig opprettede arrangementer og kommer over useriøse arrangementer samt et arrangement som har forsøkt seg på XSS. Han fjerner disse arrangementene og fratar eierne muligheten til å være arrangør i fremtiden, samt fjerner brukeren fra systemet som prøvde seg på XSS. Videre legger han til flere idrettskategorier som vil gjøre søking og sortering etter idrettstype enklere, ettersom han så flere av de nye arrangementene benyttet seg av «annet». Neste del av jobben er å svare på arrangør- og klubbsøknader, hvor han godkjenner eller aviser disse utfra reglene og hvem som egner/ikke egner seg.

Scenariet over beskriver en mulighet for hvordan systemet kan bli brukt som en *serviceadministrator*, og tester disse funksjonene for systemet:

- Fraskrive retten til å være *arrangør*/fjerne roller
- Fjerne *bruker* fra systemet
- Legge til idrettskategorier
- Godkjenne/avvise søknader
- Redigere/fjerne arrangementer

Databaseadministrator

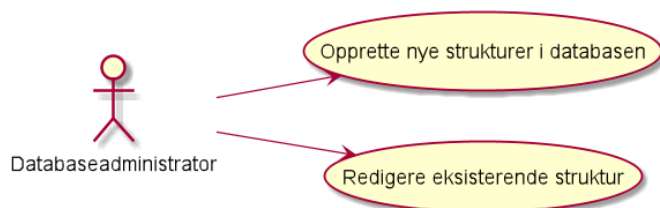
Beskrivelse

En *administrator* som korrigerer og justerer databasen for å etterfølge krav satt både til effektivitet og integritet av produktinnehavers firma.

Persona

Navn	Bjørn Rune
Alder	41 år
Yrke	<i>Databaseadministrator</i>
Utdanning	Master, Applied computer science
Interesser	Programmering, fotball og uteliv
Bruksområde i systemet	Administrator av databasen
Brukergruppe	<i>Databaseadministrator</i>

Use case diagram



Use cases

1. Som en *databaseadministrator* ønsker jeg å kunne administrere og rette opp i databasen, finjustere og bruke indekser slik at dataene blir riktige og får en akseptabel struktur.

User story

Bjørn Rune er 41 år og databaseadministrator for systemet. I løpet av arbeidsdagene bruker han tiden sin på å rette opp i feil, lage indekser og effektivisere søk i databasen.

Scenariet over beskriver en mulighet for hvordan systemet kan bli brukt som en *databaseadministrator*, og tester disse funksjonene for systemet:

- Rette opp i feil struktur eller funksjonalitet i databasen.
- Redigere databasen
- Lage indekser

Funksjonelle krav

Kravene er segmentert ut ifra hvilken modul/del av systemet kravene gjelder.

Alle kravene i kravspesifikasjonen som omhandler brukertypen *Bruker* vil også gjelde for *Klubbleder*/Gruppeleder. Både *Gruppeleder* og *Klubbleder* er fremdeles en ordinær *Bruker*, men har egne krav knyttet opp til seg i tillegg til kravene relatert til *Bruker*.

Underoverskriftene er bygd opp slik: [Brukertype.Handling.Hva som skal kunne gjøres/ikke gjøres](#)

Brukergrensesnitt

Disse kravene omhandler brukergrensesnittet og hva som skal være mulig å se i systemet.

Bruker.Brukergrensesnitt.Skal kunne se

1. Liste over alle arrangementer *brukeren* er påmeldt.
2. Liste over fremtidige arrangementer.
3. Alle tilgjengelige arrangementer.
4. Liste over fremtidige påmeldte arrangementer.
5. Liste over pågående arrangementer.
6. Liste over pågående påmeldte arrangementer.
7. All informasjon *arrangør* har oppgitt for et arrangement.
8. Resultater fra tidligere deltatte arrangementer.
9. Fornavn og etternavn til andre deltakere som har meldt seg på samme arrangement.
10. Et utsnitt av [Google Maps](#) hvor posisjonen til valgt arrangement skal vises.
11. Liste over andre deltakere som er meldt på et gitt arrangement.

Klubbleder.Brukergrensesnitt.Skal kunne se

1. Alle klubbmedlemmer i sine egne klubber.
2. Se hvilke arrangementer klubben er meldt på.
3. Tidligere arrangementer klubben har vært med på.
4. Resultatene fra tidligere arrangementer klubbmedlemmer har deltatt i.
5. En liste over status til alle medlemmer i systemet. Det vil si om medlemmene deltar i andre arrangementer med andre grupper/klubber eller individuelt.
6. Hvilket idrettslag klubben hører til.

Gruppeleder.Brukergrensesnitt.Skal kunne se

Referer til kravene for klubbleder da de i stor grad har de samme kravene. Krav som viker fra gruppeleder blir skrevet her:

1. Gruppelederen vil ikke kunne se påmeldt idrettslag da en gruppe ikke tilhører et idrettslag.

Gruppemedlem/Klubbmedlem.Brukergrensesnitt.Skal kunne se

1. En liste over andre medlemmer i gruppen/klubben en er med i.
2. Hvem som er lederen av gruppen/klubben.
3. En liste over hvilke arrangementer gruppen/klubben er meldt på.

Arrangør.Brukergrensesnitt.Skal kunne se

1. Antall påmeldte brukere i *arrangørens* egne arrangementer.
2. Antall påmeldte grupper/klubber i *arrangørens* egne arrangementer.
3. All informasjon om ens egne arrangementer.
4. Informasjon om alle påmeldte brukere i ens egne opprettede arrangementer.

Serviceadministrator.Brukergrensesnitt.Skal kunne se

1. All informasjon om alle arrangementer.
2. Alle arrangementer til enhver tid uavhengig av status.
3. Antall aktive brukere.

Status: Om et arrangement er aktiv, pågående eller fremtidig.

Registrering

Disse kravene omhandler funksjonaliteten for registrering i systemet.

Bruker.Registerer.Skal kunne

1. Registrere med følgende personalia: E-post, fornavn, etternavn, fødselsdato, ZIP-kode, Gate Adresse, kjønn og passord.
2. Velge ett passord med minimum lengde på 10 karakterer, derav minst en av disse er et tall og minst en stor bokstav.

Arrangør.Registrerer.Skal kunne

Dette er en bruker i systemet som ønsker å bli omgjort til Arrangør.

1. Registrere seg som arrangør ved å først registrere som en ordinær bruker og deretter sende søknad som går til behandling hos Serviceadministrator.

Dette er en organisasjon som ønsker å registrere som Arrangør.

2. Registrere seg direkte som en *arrangør*. Dette vil si at en organisasjon skal kunne registrere seg i systemet uten å sende søknad slik en ordinær *bruker* må utføre. (Dette fordi organisasjoner automatisk verifiseres av systemet utfra gitt informasjon under registreringen).

3. Registrere med følgende personalia: Navnet til *organisasjonen*, ni-siffer organisasjonsnummer, e-postadresse og passord.

Systemadministrator.Registerer.Skal ikke kunne

1. Registrere seg i systemet på samme måte som Bruker og Arrangør. Systemadministrator skal bli valgt ved oppstart av systemets eksistens.

Innlogging

Disse kravene omhandler funksjonaliteten for innlogging i systemet.

Alle brukere.Innlogging.Skal kunne logge inn

1. Ved å bruke en kombinasjon av e-post og passord.
2. Med [tredjepartsintegrasjoner](#) som Google og Facebook.

Sortering

Disse kravene omhandler sortering og søk av arrangementer i systemet.

Alle brukere.Sorterer arrangementer.Skal kunne

1. Sortere alle arrangementer etter stigende dato.
2. Sortere alle arrangementer etter synkende dato.
3. Sortere alle arrangementer i alfabetisk rekkefølge, høyest og lavest i henhold til selve kolonnen.
4. Filtrere søkene ut ifra ønsket aldersgrense på arrangementet.
5. Filtrere arrangementer ut ifra valgt idrett.
6. Filtrere etter valgt idrett på arrangementet.

Alle brukere.Søker arrangement.Skal kunne søke arrangementer

1. Med et arrangements navn.
2. I nærheten av en gitt ZIP kode.
3. Opprettet av en gitt *arrangør*.
4. Som starter en gitt dag.
5. Som er innenfor et datointervall.
6. Med et gitt stedsnavn.

Meldinger

Disse kravene omhandler meldingsfunksjonalitet i systemet.

Alle brukere.Opprette meldingsgruppe.Skal kunne

1. Legge til ønskede medlemmer i meldingsgruppen.
2. Sette andre medlemmer av meldingsgruppen som administrator av gruppen ved opprettelse.

Alle brukere. Sender melding til meldingsgruppe.Skal kunne

1. Sende fellesmelding til alle medlemmer av meldingsgruppen.
2. Sende melding til enkeltmedlemmer.

3. Opprette en melding, deretter sette en valgt dato om når meldingen skal bli sendt ut til deltakere i meldingsgruppen.
4. Send melding til enkeltmedlemmer av meldingsgruppen.

Disse kravene omhandler en Bruker som allerede har opprettet en meldingsgruppe. Det kan også være en bruker som har blitt satt som administrator av meldingsgruppeoppreteren.

Alle brukere.Administrere meldingsgruppe.Skal kunne

1. Fjerne deltakere i meldingsgruppen.
2. Legge til deltakere i meldingsgruppen.
3. Sette rollen til ønskede deltakere som *administrator av gruppen*.
4. Slette meldinger i meldingsgruppen.
5. Ekskludere enkeltmedlemmer i meldingsgruppen fra å motta fellesmeldinger.

Arrangør.Sender melding.Skal kunne

1. Send fellesmelding til alle som er påmeldt et gitt Arrangement (Gjelder både Gruppe/Klubb og enkeltbrukere).

Disse kravene forteller om rettigheter hver enkelt bruker har til å sende meldinger til andre brukergrupper.

Bruker.Sender melding.Skal kunne

1. Send melding til disse brukergruppene: *Bruker, Klubbleder og Gruppeleder*.

Arrangør.Sender medling.Skal kunne

1. Send melding til disse brukergruppene: *Bruker, Klubbleder, Gruppeleder og Serviceadministrator*.

Klubbleder/Gruppeleder.Sender melding.Skal kunne

1. Send melding til disse brukergruppene: *Bruker, Arrangør, Klubbleder, Gruppeleder og Serviceadministrator*.

Serviceadministrator.Sender melding.Skal kunne

1. Send melding til disse brukergruppene: *Bruker, Arrangør, Klubbleder og Gruppeleder*.

Brukerspesifikke krav

Disse kravene er segmentert ut ifra hvem som utfører handlingen. De er formatert slik da kravene kun forekommer hos en enkelt *bruker* i systemet.

Bruker

Dette er krav i sammenheng med brukeren *bruker*.

Bruker.Påmelde arrangement.Skal kunne

1. Melde seg på flere arrangementer.
2. Betale for å melde seg på et arrangement.
3. Dele det påmeldte arrangementet i sosiale medier via delingsknapper i systemet.

Bruker.Påmelde arrangement.Skal ikke kunne melde på arrangementer

1. Som sammenfaller med et annet allerede påmeldt arrangement.
2. Hvor *brukeren* ikke oppfyller alderskravet.
3. Som har sin sluttdato før dagens dato
4. Der *brukeren* allerede er påmeldt.
5. Som sammenfaller med arrangementer en gruppe/klubb *brukeren* er deltaker i.

Bruker.Avmelde arrangement.Skal kunne melde av arrangementer

1. Hvor det allerede har blitt betalt (refundering er ikke mulig).
2. Hvor det ikke er betaling.
 1. Som har utgått.
 2. Som pågår.
 3. Som *brukeren* allerede har meldt av.

Bruker.Opprette gruppe.Skal kunne

1. Sende invitasjon til fremtidige *gruppemedlemmer* gitt en e-post. Dette uavhengig om eposten er registrert i systemet.
2. Sende invitasjon til allerede registrerte brukere i systemet.

Bruker.Behandle invitasjon.Skal kunne

1. Godta invitasjoner til klubb/gruppe.
2. Avslå invitasjoner til klubb/gruppe

Klubbleder

Dette er krav som har en sammenheng med *brukeren* [klubbleder](#).

Klubbleder.Administrerer klubb.Skal kunne

1. Invitere nye medlemmer.
2. Fjerne eksisterende *medlemmer*.
3. Slette klubben.
4. Endre navnet på klubben en gang i året.
5. Gi ønskede medlemmer i klubben *administrerende rettigheter*.

Administrerende rettigheter: Brukeren som blir tildelt denne rollen vil nå ha de samme rettigheten som klubblederen som opprettet klubben.

Klubbleder.Administrerer klubb.Skal ikke kunne

1. Slette klubb dersom klubben er meldt på arrangementer innenfor sju dager.
2. Slette klubb dersom klubben har medlemmer som er aktive i et pågående arrangement.

Klubbleder.Påmelder klubb.Skal kunne

1. Melde alle *medlemmer* på et arrangement. Her vil *medlemmene* selv bli spurt om å verifisere at de ønsker å delta på arrangementet.
2. Melde på enkelt medlemmer i klubben på arrangementer dersom medlemmet godtar påmeldingen.

Klubbleder.Påmelder klubb.Skal ikke kunne

1. Melde på *klubbmedlemmer* i et tidsrom hvor vedkommende allerede er påmeldt et arrangement.

Klubbleder.Melder klubb av arrangement.Skal kunne

1. Melde seg av alle påmeldte arrangementer.
2. Melde enkeltmedlemmer av arrangementer.
3. Melde av klubben, men la utvalgte medlemmer gjenstå.

Klubbleder.Melder klubb av arrangement.Skal ikke kunne

1. Melde klubben av arrangementer som allerede har blitt arrangert.
2. Melde klubben av arrangementer som pågår.

Gruppeleder

Dette er krav som har en sammenheng med *brukeren* [Gruppeleder](#).

Denne brukergruppen har de samme kravene som brukergruppen *Klubbleder*. Krav som avviker fra *Klubbleder*, er nevnt her:

Gruppeleder.Søker som klubbleder.Skal kunne

1. Søke om gruppen kan bli en autorisert klubb ved å oppgi følgende informasjon: Tilhørende idrettsorganisasjon og en kommentar dersom ønsket.

Klubbmedlem/gruppedlem

Dette er krav som har en sammenheng med brukerne [klubbmedlem/gruppedlem](#).

Medlem.Administrere.Skal kunne

1. Melde seg ut av klubben/gruppen.
2. Melde seg ut av arrangementer gruppen/klubben har meldt seg på.
3. Melde seg på et arrangement individuelt om klubben/gruppen har meldt seg ut av et arrangement dersom arrangementet har dette som en mulighet.
4. Være med i flere klubber/grupper.

Medlem.Administrere medlemskap.Skal ikke kunne

1. Melde seg ut av klubben/gruppen dersom gruppen har et pågående arrangement hvor klubben/gruppen er påmeldt.

Arrangør

Dette er krav som har en sammenheng med *brukeren* [Arrangør](#).

Arrangør.Opprette arrangement.Skal kunne

1. Opprette arrangementer.
2. Legge til grafiske elementer, bilder og/eller video i beskrivelsen av arrangementet.
3. Bestemme lovlig intervall for alder, inklusivt minimum og maksimum, av deltakere.
4. Velge om arrangementet er gratis eller krever en betaling.

5. Benytte [Klarna](#) som betalingsløsning dersom arrangementet krever betaling.

Arrangør.Opprette arrangement.Må velge

Følgende informasjon ved opprettelse av arrangement:

1. Navn på arrangementet.
2. Hvilken idrett arrangementet skal ha.
3. Adressen hvor arrangementet skal bli holdt.
4. Maks antall plasser.
5. Dato når arrangementet starter.
6. Dato når arrangementet slutter.
7. Klokkeslett når arrangementet starter.
8. Klokkeslett når arrangementet slutter.
9. Om arrangementet krever betaling for påmelding.
10. Om arrangementet krever betaling skal arrangør kunne velge prisen.
11. Om arrangementet skal benytte ID-brikker for tidtaking og målplassering. Hvis arrangør ikke velger ID-brikker, må resultater skrives inn manuelt.
12. En beskrivelse av arrangementet med valgfri tekst. Dette vil typisk være hvor langt(km) løpet/rittet/rennet vil være og en generelt beskrivende tekst om arrangementet.
13. Et aldersintervall med satt minstealder og eldstealder.

Arrangør.Opprette arrangement.Skal ikke kunne

1. Opprette arrangementer med støtende navn.

Arrangør.Administrere arrangement.Skal kunne

1. Endre all informasjon i sine egne arrangementer frem til 4 uker før arrangementet skal bli holdt.
2. Utestenge brukere, klubber og grupper fra sine egne arrangementer.
3. Slette sine egne arrangementer frem til fire uker før arrangementet skal bli holdt.
4. Endre aldersintervallen.

All informasjon: Navnet på arrangementet, antall deltakere, idrett, maks antall deltakere, adresse, dato, navn og alder til påmeldte brukere.

Arrangør.Administrere arrangement.Skal ikke kunne

1. Endre arrangementets navn til et støtende navn.
2. Slette sine egne arrangementer etter det er fire uker til arrangementet skal bli holdt.
3. Slette utgåtte arrangementer.
4. Slette andre arrangørers arrangementer.

Administratorer

Dette er krav som har en sammenheng med brukertypene [serviceadministrator](#) og [databaseadministrator](#).

Serviceadministrator.Skal kunne

1. Legge til nye arrangement-kategorier/idretter.
2. Redigere alle arrangementer.
3. Slette alle arrangementer.
4. Godkjenne arrangørsøknader fra *brukere*.
5. Avslå arrangørsøknader fra *brukere*.
6. Utestenge alle *Brukere* fra systemet.
7. Godkjenne klubbsøknader fra grupper.
8. Avslå klubbsøknader fra grupper.
9. Slette klubber.
10. Slette grupper.

Serviceadministrator.Skal ikke kunne

1. Opprette egne arrangementer.
2. Utestenge andre serviceadministratorer fra systemet.

Databaseadministrator.Skal kunne

1. Rette opp i feil i databasen.
2. Legge til ny struktur i databasen.
3. Redigere allerede eksisterende struktur i databasen.
4. Ha full tilgang til selve databasesystemet. Derav muligheten til å regulere selve DBMS.

Systemkrav

Dette er funksjonelle krav systemet skal utføre automatisk som en følge av utførte handlinger. De er segmentert i henhold til hvem/hva systemkravene er relatert til.

System.Innlogging.Skal

1. Vise følgende advarsel for *bruker* dersom feil epost/passord er tastet inn: Enten epost og/eller passord er feil, vennligst prøv igjen. Her skal det ikke vises mer informasjon enn dette. Definisjonen av ugyldig informasjon er gitt her.

System.Arrangør.Skal

1. Bruke Google Maps for å kunne finne lokasjon til arrangement basert på adressen *arrangør* har oppgitt som deretter blir vist i brukergrensesnittet.
2. Send en notifikasjon og e-post til *arrangør* dersom en feil skulle skje med lasting av lokasjon hvor *arrangøren* skal få beskjed om å plassere stedslokasjonen selv.
3. Ved invalid input i datafelter skal systemet varsle *arrangør* om dette med hvilke data som er feil.
4. Validere at gitt alder er større enn eller lik 18 ved opprettelse eller søknad som arrangør.
5. Validere at *bruker* allerede eksisterer i systemet.
6. Send et utfylt skjema som er valid, til en *serviceadministrator*, samt en push melding til *serviceadministratoren*.

7. Ved godkjent søknad om å bli *arrangør*, sende push melding til gitte *bruker*, samt mail, om dette.
8. Dersom en organisasjon ønsker å registrere seg som *arrangør* skal dette skje automatisk uten behov for manuell vurdering, ved sjekking mot et statlig register for organisasjoner.

System.Bruker.Skal

1. Generere en konfirmasjonslink som blir sendt til *brukerens* e-post ved brukerregistrering.
2. Legge til *bruker* i systemet dersom *brukeren* godtar konfirmasjonen.
3. Gjøre om forbokstaver på fornavn og etternavn dersom bokstavene *bruker* skrev inn ikke er store.
4. Generere en kvittering om godkjent kjøp til *brukeren* ved kjøp av billett til et arrangement.
5. Sende resultater fra arrangementer en *Bruker* har deltatt i videre til lagring i database. Dette skjer ved at ID-brikker automatisk sender inn resultatene, eller ved at arrangør fører inn manuelt.
6. Gi beskjed til *brukeren* om innloggingsinformasjon ikke er gyldig. Meldingen skal forklare hva som ikke er gyldig.
7. Sende notifikasjon og e-post når det forekommer endringer i en *brukers* påmeldte arrangement.
8. Sende notifikasjon og e-post til alle *brukere* som er påmeldt et arrangement når det er 24-timer igjen til oppstart.
9. Sende e-post til *bruker* med bekreftelse om avmelding av et arrangement. E-posten skal inneholde navn på arrangement, startdato og sluttdato.
9. Sende kvittering på e-post ved bekreftet betaling (systemet vil motta en respons fra *klarna*, som systemet deretter videresender til *bruker*).
10. Sende billett til arrangement på e-post om *bruker* har betalt og meldt seg på et arrangement.
11. Sende notifikasjon og e-post når en *Bruker* blir lagt til i en meldingsgruppe.
12. Sende notifikasjon og e-post ved uleste meldinger når melding ikke har blitt åpnet på over en uke.
13. En link skal sendes til *person* om gruppe/klubbleder inviterer person ved hjelp av e-post. Denne linken skal føre til

Person: Dette et vilkårlig menneske som end ikke har registrert seg i systemet.

System.Bruker.Skal ikke

1. Legge til *bruker* i systemet dersom *brukeren* aldri godtar konfirmasjonen.
2. Gi refusjon til *brukere* som melder seg av allerede betalte arrangementer (angrerettloven kapittel 5).

Konfirmasjonen: Dette er e-posten en bruker mottar ved registrering i systemet. Brukeren må bekrefte denne for å bli lagt til i systemet.

System.Gruppelder.Skal

1. Sette rollen som *gruppeleder* på den *brukeren* som opprettet gruppen.
2. Bytte status fra *gruppeleder* til *klubbleder* ved godkjent klubb søknad.
3. Sende e-post og notifikasjon når gruppen har fått behandlet sin søknad som klubb.

System.Klubbleder.Skal

1. Sette rollen som *klubbleder* på den *brukeren* som oppretter en klubb.
2. Sende notifikasjon og e-post til *klubbleder* dersom *klubbmedlemmer* forlater klubben.
3. Sende notifikasjon og e-post når *klubbleder* mottar melding fra *gruppemedlemmer*.
4. Sende notifikasjon og e-post når *klubbleder* har mer enn en uke gamle uleste varsler fra *gruppemedlemmer*.

System.Klubbmedlem/Gruppemedlem.Skal

1. Sende notifikasjon og e-post når *brukeren* mottar melding fra *klubbleder* eller gruppeleder.
2. Sende notifikasjon og e-post når *klubbleder* melder *klubben* på et arrangement.
3. Sende notifikasjon og e-post når *klubbleder* melder *klubben* av et arrangement.
4. Sende notifikasjon og e-post når *klubbleder* fjerner *medlemmet* fra klubben.

System.Serviceadministrator.Skal

1. Generere en liste med pågående arrangementer per dag og sende beskjen til *serviceadministrator*.
2. Sende notifikasjon og e-post til *serviceadministrator* når en *arrangør* blir registrert i systemet.

System.Gruppe/Klubb opprettelse.Skal

1. Lage en meldingsgruppe bestående av medlemmer i Gruppen/Klubben ved opprettelse.

Ikke funksjonelle krav

System

1. Skal lagre kryptert brukerinformasjon.
1. Skal lagre resultatinformasjon fra arrangementer.
2. Resultatinformasjon skal kunne innhentes og lagres automatisk dersom *arrangør* bruker ID-brikker under arrangementer.
3. Lagre informasjon i database om hvilken *serviceadministrator* som godkjente en *arrangørsøknad*, samt *arrangør* og tidspunktet dette ble utført på.
4. Lagre informasjon i database om hvilken *serviceadministrator* som godkjente en *klubbsøknad*, samt *klubbleder* og tidspunktet dette ble utført på.

Resultatinformasjon: Dette er informasjon som blir lagret i systemet etter at deltakere har utført aktivitetene sine (Brukt tid og tilhørende arrangement).

Databaserelaterte krav

1. Databasen skal ta imot all data som systemet sender til databasen.
2. All data i systemet skal ha en backup.
3. Backup skal utføres automatisk hver time.
4. Backup skal slettets automatisk hver fjerde dag.
5. All data må ha et presisert og dokumentert formål for innsamling.

6. Dataintegriteten skal opprettholdes til enhver tid.
7. Dersom data ikke har vært i bruk på over 5 år, skal den automatisk bli slettet fra systemet. Dette gjelder data om både *brukere* og *arrangementer*.

Ytelsesrelaterte krav

1. Systemet må kunne være tilgjengelig mellom 04.00 – 00.00, hver dag.
2. Responstiden for alle *brukere* mot systemet skal maksimalt være 8 sekunder.
3. Responstiden for uthenting av data i systemet skal maksimalt være 5 sekunder.

Sikkerhetsrelaterte krav

1. Ved registrering av data skal systemet kunne sjekke etter validiteten på dette.
2. Passord og sensitiv informasjon skal krypteres slik at ingen passord eller sensitiv informasjon sendes eller lagres i klartekst.
3. Systemet skal ta hensyn til norske lover samt GDPR lovgivning. Dette gjelder primært oppbevaring og bruk av brukerdata, samt annen personlig og sensitiv informasjon.
4. Alle Passord enten ved opprettelse eller ved autentisering av *bruker*, skal krypteres med en salt hos klient før det sendes videre inn mot systemet.

Gjenopprettelserelaterte krav

1. Ved en fatal feil i systemet skal containeren med feilen slås av og oppgaver skal bli delegert til andre fungerende containere.
2. Ved en nedetid for hele av systemet på tvers av containere skal utviklere bli tilkalt for å fikse dette på maksimalt to timer.

Safety relaterte krav

1. Personaldata skal kun være tilgjengelig for *brukeren* det omhandler og for *serviceadministrator* (ved behov).

Redundansrelaterte krav

1. Systemet skal operere med flere instanser av samme funksjonalitet for å hindre nedetid.

Tilgjengelighetskrav

Dette er de mest relevante kravene hentet fra WCAG 2.0 – standarden i sammenheng med dette systemet. Flere WCAG 2.0 krav finnes [her](#).

Lovpålagt

1. Kontrastforhold på fargenivå til bakgrunn og tekst. Tekstlig innhold skal ha et kontrastnivå på minst 4.5:1 i forhold til bakgrunnen. Større tekstlig innhold som overskrifter og underoverskrifter skal ha et kontrastforhold på minst 3:1.
2. Formidling av informasjon skal ikke basere seg kun på farger. Andre visuelle virkemidler må brukes for å formidle ønsket informasjon.
3. Alle funksjoner skal være tilgjengelig ved å kun bruke tastatur. Det skal være mulig å tabbe gjennom innhold som er repetitivt for hver nettside som lastes på nettstedet.
4. Tekstlig alternativer for innhold som består av lyd, bilder og video.
5. Ved bruk av tastatur skal fokusert innhold være markert for *brukeren*.
6. Tekstlig innhold skal kunne zoomes inn opptil 200% uten at det påvirker funksjonalitet eller lesbarhet.

Ikke lovpålagt

1. Optimale krav for kontrast er 7:1 dersom det er mulig å utføre.
2. Tekstlig innhold burde ikke bli fremstilt i form av bilder.
3. Brukergrensesnittet skal være intuitivt og enkelt å bruke slik at selv de med lav datakyndighet enkelt kan bruke systemet.

Estimering og prioritering av funksjonalitet

Prioriteringer av funksjoner i systemet blir representert ut ifra et hierarki av disse gruppene:

Small(S), medium(M), large(L) og X-large(XL).

Small er minsteverdi og X-large er største verdi.

Brukerfunksjoner

Funksjon	Utviklingstid	Viktighet
Betale for å melde seg på arrangement	L	L
Melde på arrangement	S	L
Se liste over pågående/fremtidige arrangementer	S	L
Påmeldingsrestriksjoner (alderskrav, utgått dato, allerede påmeldt, sammenfaller med andre arrangementer,	S	L
Avmeldingsrestriksjoner (utgåtte, pågående og allerede deltatte arrangementer)	S	L
Registrering av <i>bruker</i>	M	M
Se resultater (tid brukt) fra tidligere deltatte arrangementer.	M	M
Innlogging	M	M
Melde av arrangementer	S	M
Se informasjon om arrangementer	S	M
Se liste over alle påmeldte arrangementer	S	M
Opprette gruppe	L	S
Se utsnitt av Google Maps for arrangementer	L	S
Innlogging med tredjepartsintegrasjon (Google,Facebook)	L	S

Meldingsfunksjonalitet	L	S
Sortere arrangementer	M	S
Søke etter arrangementer.	M	S

Klubb/Gruppe -lederfunksjoner

Funksjon	Utviklingstid	Viktighet
Legge til <i>medlemmer</i>	L	M
Melde klubb på arrangement	M	M
Se oversikt over alle <i>medlemmer</i>	S	M
Se oversikt over påmeldte arrangementer	S	M
Melde klubb av et arrangement	L	S
Meldingsfunksjonalitet	L	S
Fjerne <i>medlemmer</i>	M	S
Gi beskjed til <i>enkeltmedlemmer</i>	M	S
Gi fellesbeskjeder til hele gruppen	M	S
<i>Gruppeleder</i> kan søke om å bli <i>klubbleder</i>	S	S
Se oversikt over tidligere deltatte arrangementer	S	S

Klubb/Gruppe – medlemsfunksjoner

Funksjon	Utviklingstid	Viktighet
Melde seg ut av gruppe/klubb	S	M
Meldingsfunksjonalitet	L	S
Melde seg ut restriksjoner (om gruppen/klubben er meldt opp til et allerede pågående arrangement)	S	S
Melde seg selv av arrangement gruppe/klubb allerede er påmeldt	S	S
Individuelt melde seg på et arrangement klubb/gruppe har meldt seg ut av	S	S

Arrangørfunksjoner

Funksjon	Utviklingstid	Viktighet
Registrere som <i>arrangør</i> . (både <i>brukere</i> og <i>organisasjoner</i>)	XL	XL
Opprette arrangement.	L	XL
Kreve betaling for arrangementer.	M	L
Redigere opprettede arrangementer	S	L
Sende pushfunksjoner til påmeldte <i>brukere</i>	L	M
Fjerne opprettede arrangementer.	S	M
Se informasjon om deltakere i arrangementene sine	S	M
Fjerne eget arrangement	S	M
Legge til video/bilder og andre medietyper rettet mot arrangementet	M	S

Meldingsfunksjonalitet	L	S
------------------------	---	---

Serviceadministratorfunksjoner

Funksjon	Utviklingstid	Viktighet
Vurdere å Godkjenne/avslå arrangørsøknader	S	XL
Redigere arrangementer	M	L
Slette arrangementer	M	L
Meldingsfunksjonalitet	L	S
Legge til nye arrangement-kategorier	S	M

Systemfunksjoner

Funksjon	Utviklingstid	Viktighet
Sende brukerinformasjon til lagring i database	M	XL
Legge til <i>bruker</i> i systemet dersom konfirmasjonslink er godkjent	S	XL
System skal sette rollen <i>gruppeleder</i> på <i>bruker</i> som oppretter gruppe	S	L
System skal endre rolle fra <i>gruppeleder</i> til <i>klubbleder</i> når <i>søknad</i> er godkjent	S	L
System skal sette rollen <i>klubbleder</i> på <i>bruker</i> som oppretter klubb	S	L
Epost til <i>bruker</i> med avmeldingsbekreftelse	M	M
Generere konfirmasjonslinker	M	M
Generere kvittering for kjøp av billett	M	M
Advarsel ved feil input	S	M
Notifikasjon/epost for endringer i <i>brukers</i> arrangement	S	M
Notifikasjon/epost når klubbsøknad er behandlet	S	M
Notifikasjon/epost når <i>klubbleder</i> mottar melding fra <i>gruppemedlem</i>	S	M
Notifikasjon/epost når <i>bruker</i> mottar melding	S	M
Notifikasjon/epost til <i>medlem</i> når <i>klubbleder/gruppeleder</i> melder <i>bruker</i> av arrangement	S	M
Notifikasjon/epost til <i>medlem</i> når <i>klubbleder/gruppeleder</i> fjerner <i>bruker</i> fra klubb	S	M
Notifikasjon/epost til <i>serviceadministrator</i> når <i>arrangør</i> er registrert	S	M
Google Maps finner lokasjon basert på adresse	L	S

Opprette meldingsgrupper som en følge av opprettelse av klubb/gruppe	L	S
Advarsel til <i>arrangør</i> om Google ikke finner lokasjon	S	S
Gjøre om forbokstav i navn til stor dersom den er liten	S	S
Sende resultat fra arrangement til database	S	S
Notifikasjon/epost til <i>brukere</i> påmeldt arrangement 24 før oppstart	S	S
Notifikasjon/epost dersom <i>klubbmedlem</i> forlater klubb	S	S
Notifikasjon/epost når <i>klubbleder</i> har mer enn 1 uke gamle uleste varsler	S	S
System skal lagre informasjon i database angående hvilken <i>serviceadministrator</i> som godkjente arrangørsøknad	S	S
System skal generere liste med pågående arrangementer per dag, og sendes til <i>serviceadministrator</i>	S	S

Tredjepartsintegrasjoner

Dette er konkret funksjonalitet som utføres av en tredjepart. Disse integreringene er valgt ut for å forenkle kompleksiteten av systemet, og skape bedre brukeropplevelse. Krav som tilhører integrasjoner vil bli referert til.

Betaling

Klarna

Klarna er en betalingsløsning som tilbyr elektronisk betaling. Tilbyderen inkluderer muligheter som faktura og delbetaling. Dermed vil kundene ha mulighet til å velge passende alternativ.

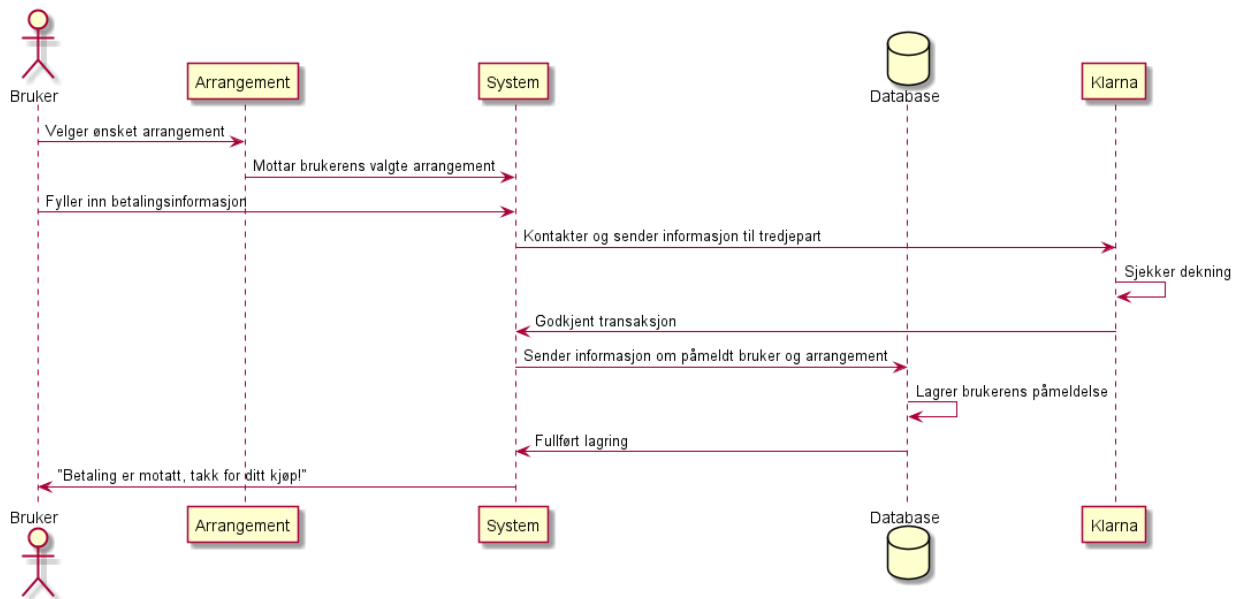
Klarna vil bli integrert i systemet for å ta seg av transaksjoner som omgår kjøp av alt fra billett kjøp, drakter og annet utstyr som trengs til arrangementer, eller abonnement for å kunne benytte systemet dersom tjenesten ikke skal være gratis.

For å integrere systemet med Klarna må Merchant Portal benyttes. Dette er et av Klarnas ordrehåndteringssystemer for nettbutikker med en Klarna-avtale. Deretter hentes API-detaljene herfra og legges inn på plattformen for å kunne ta imot kjøp.

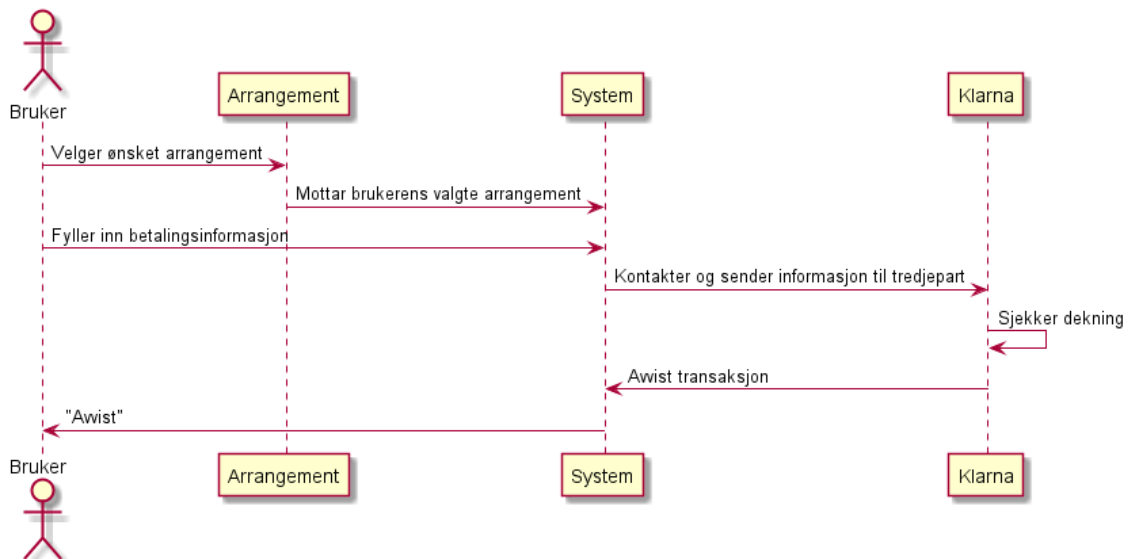
Tilhørende krav:

Link til segment	Valgte krav (Gitt med punkt nr)
System.Bruker.Skal	8

Sekvensdiagrammet viser flyt ved godkjent betaling og påmelding av arrangement.



Sekvensdiagrammet viser flyt ved avvist oppkobling eller transaksjon mot Klarna.

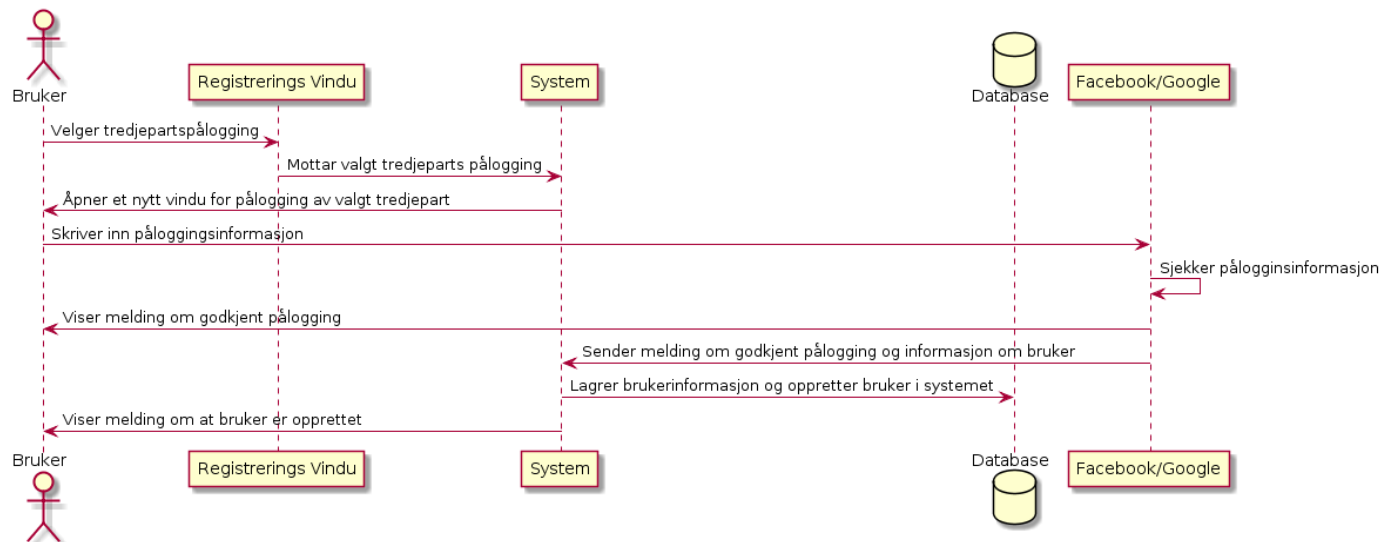


Innlogging

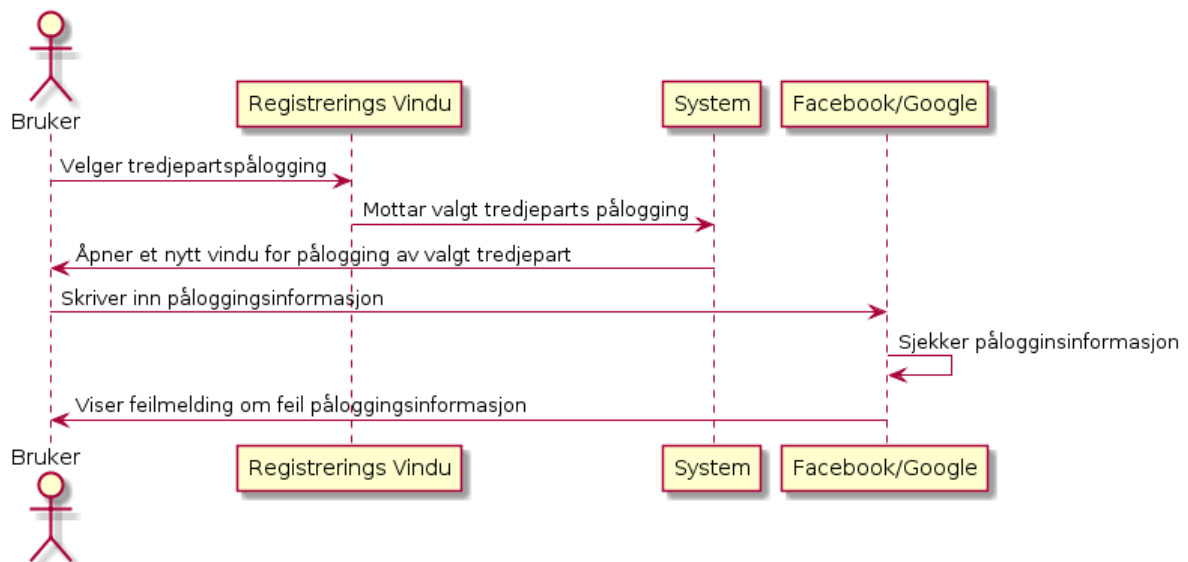
Facebook og Google

Innlogging via tredjeparter vil være et alternativ for brukere av systemet for å forenkle registrering- og innloggingsprosessen. I tillegg er det en god måte å samle data om brukerbasen av systemet. Når brukere benytter tredjepartsinnlogging, vil de legge igjen all offentlig informasjon om seg selv. I senere tid vil dataen om brukerne kunne benyttes for å tilpasse ut ifra målgrupper som benytter systemet.

Sekvensdiagrammer viser flyt ved godkjent pålogging ved bruk av Facebook eller Google



Sekvensvensdiagrammet viser flyt ved avvist pålogging ved bruk av Facebook eller Google



Tilhørende krav:

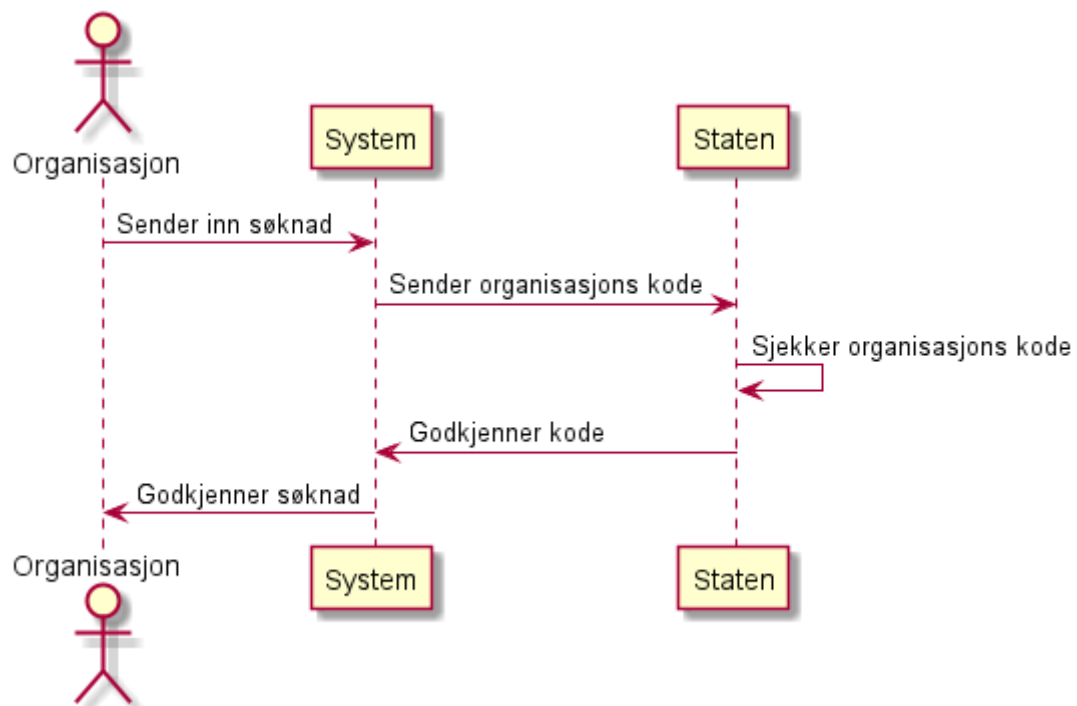
Link til segment	Valgte krav (Gitt med punkt nr)
Alle brukere.Innlogging.Skal kunne logge inn	2

Andre tredjepartsintegrasjoner

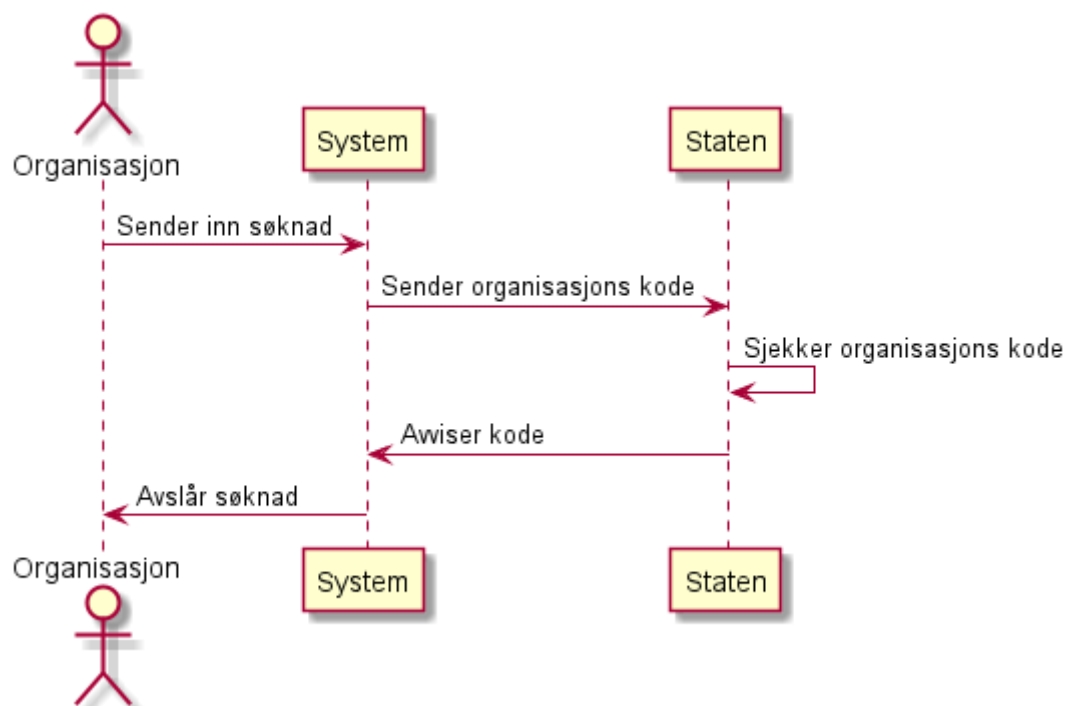
Statens organisasjonsregister

Statens organisasjonsregister blir brukt for å hente data om registrerte organisasjoner i Norge. Dette er for å kunne validere organisasjoner som ønsker å registrere seg i systemet er reelle organisasjoner.

Sekvensensdiagrammet viser flyt ved godkjent søknad om automatisk registrering som arrangør



Sekvensensdiagrammet viser flyt ved avvist søknad om automatisk registrering som arrangør

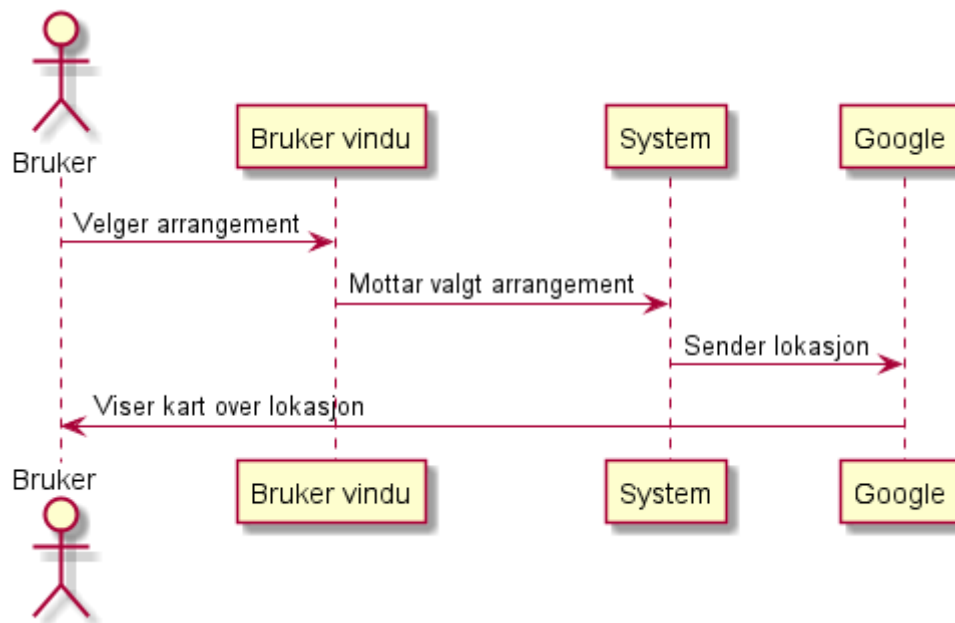


Tilhørende krav:

Link til segment	Valgte krav (Gitt med punkt nr)
Arrangør.Registrerer.Skal kunne	1, 2

Google maps

Google Maps blir brukt for å vise et kart med gitt lokasjon fra et arrangement i brukergrensesnittet. Hensikten med denne integreringen er å enkelt kunne fremvise lokasjonen til arrangementer.



Tilhørende krav:

Link til segment	Valgte krav (Gitt med punkt nr)
System.Arrangør.Skal	1

ID-brikke

Systemet vil integrere en ID-brikke som gjør at *arrangører* kan ta seg nytte av automatisk innhenting av resultater(tid) knyttet til løp/skirenn/sykkelritt. ID-brikkene vil fungere slik at de alle starter å telle når løpet starter, og stoppes når brikken passerer målstreken. Ettersom hver deltager har sin ID-brikke registrert til sitt navn, vil resultatet automatisk kunne hentes ut og legges til i systemet til riktig *bruker*.

Tilhørende krav:

Link til segment	Valgte krav (Gitt med punkt nr)
System.Bruker.Skal	5
Bruker.Brukergrensesnitt.Skal kunne se	7
Klubbleder.Brukergrensesnitt.Skal kunne se	4

Arkitektur og sammenhenger i systemet

Introduksjon

En overordnet beskrivelse om hva de forskjellige partene i systemet er og hvordan de kommuniserer.

Systemet består av fem hoveddeler:

1. Front end
2. Back end – Distribusjon av oppgaver
3. Back end – Logikk og funksjonalitet

4. Database – Lagring av arrangement data
5. Database – Lagring av brukerinformasjon

Beskrivelse av partene

Front end

Designet/brukergrensesnittet og den delen av systemet hvor oppførselen til systemet baserer seg på *brukers* valg og handlinger utført gjennom et webgrensesnitt. Inkluderer alle funksjonelle krav de forskjellige *brukerne* i systemet kan utføre.

Back end – Distribusjon av oppgaver

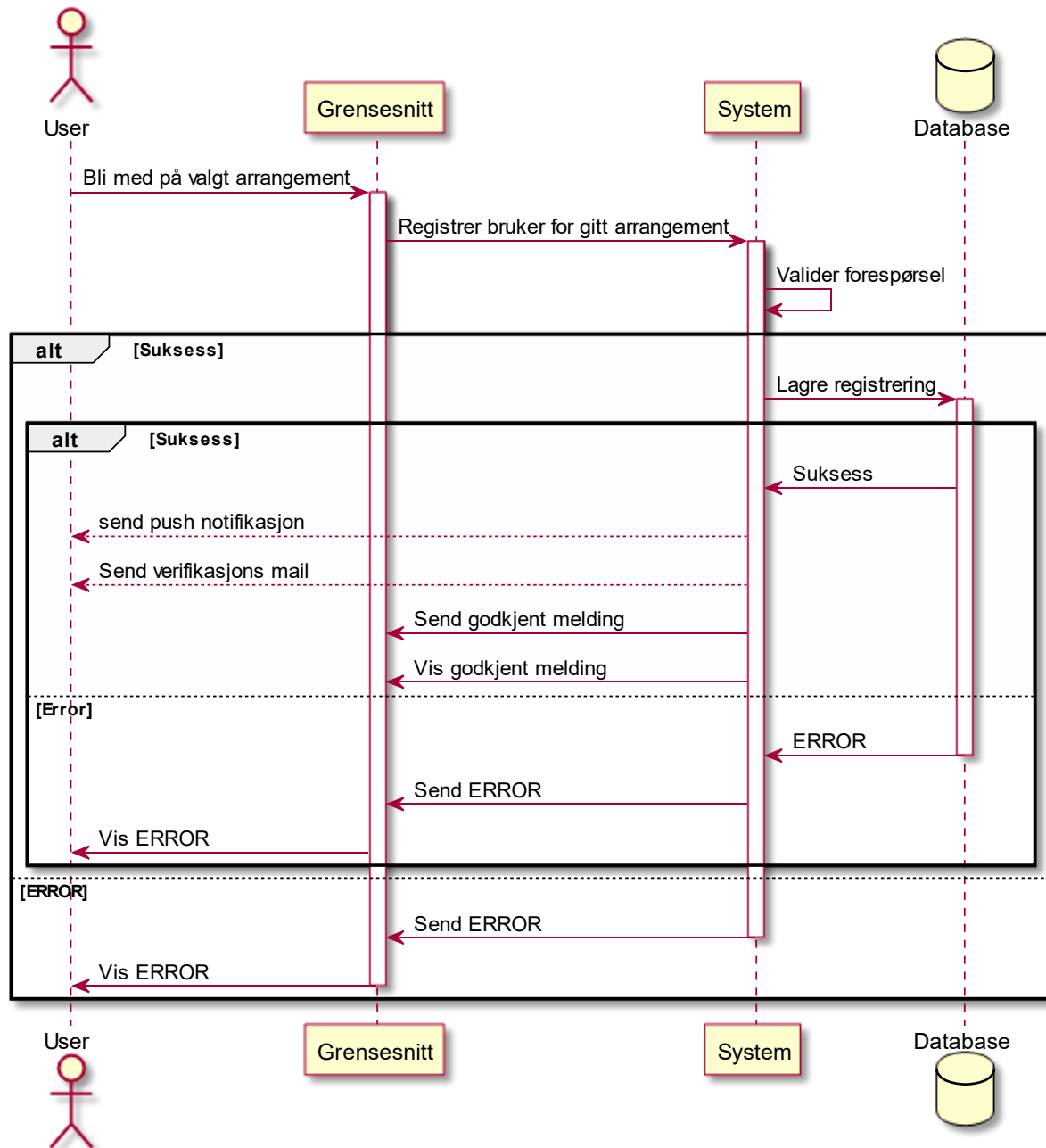
Denne parten har til oppgave å organisere og administrere innpakkede containers, som inneholder primær logikken for systemet.

Back end – Logikk og funksjonalitet

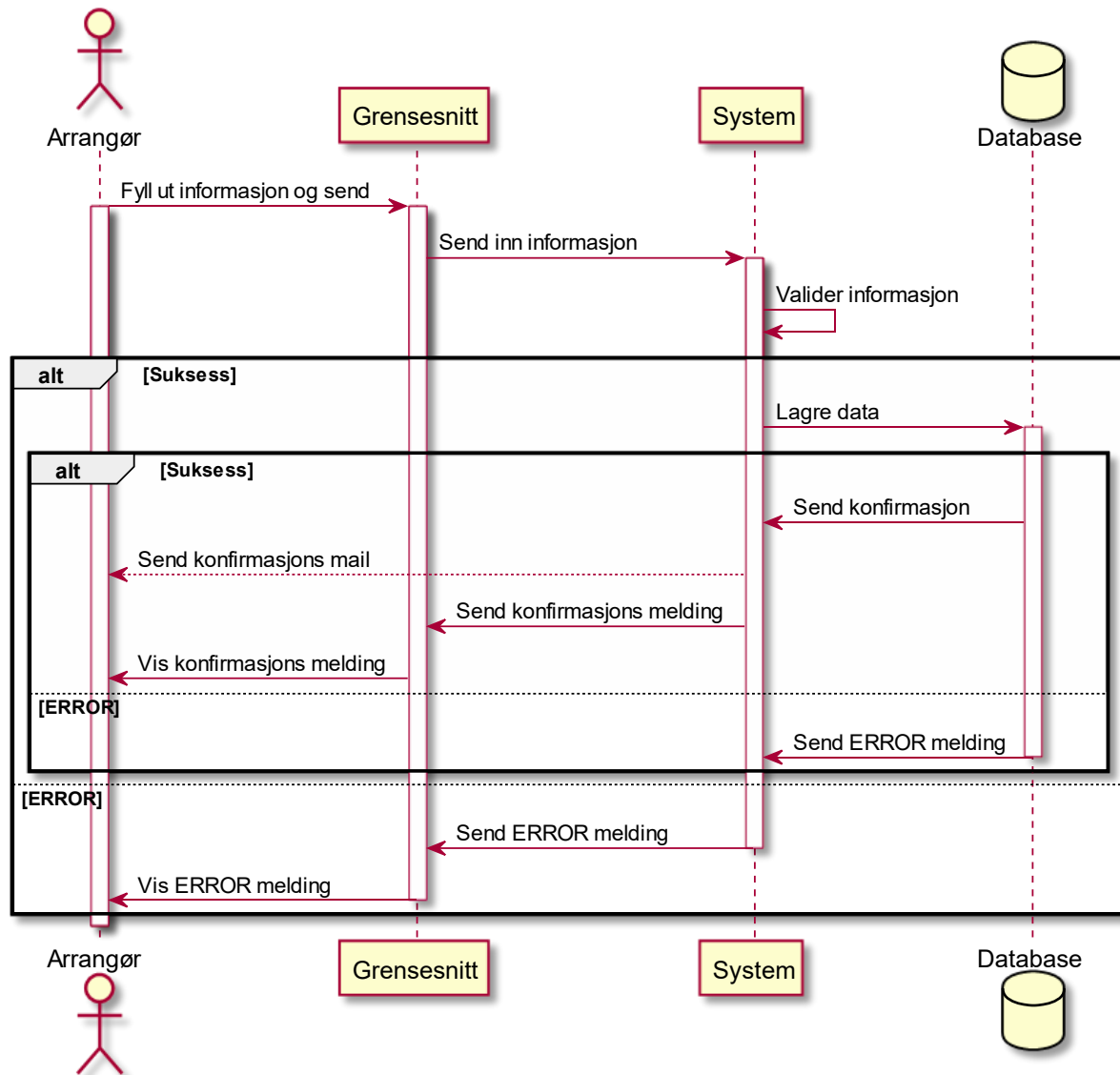
Dette er applikasjonens primære logikk. Det er denne modulen som har hovedansvaret for behandling av data samt den faktiske kjernefunksjonaliteten. Dermed utgjør dette innholdet nevnt i modulen for distribusjon av oppgaver.

Følgelig innebærer dette funksjonaliteter som endelig verifisering av *bruker* input, behandling av forespørsler opp mot database og annen logisk funksjonalitet

Sekvensdiagrammene viser antatt flyten i system gitt beskrivelsen av arkitekturen og kravspesifikasjonen. Det er ekskludert for alle tredjeparts integrasjoner for å lette på potensielt støy i diagrammet.



Figur 1 Viser flyten i systemet ved registrering for arrangementet



Figur 2 Viser hvordan flyten mellom de ulike modulene er i systemet

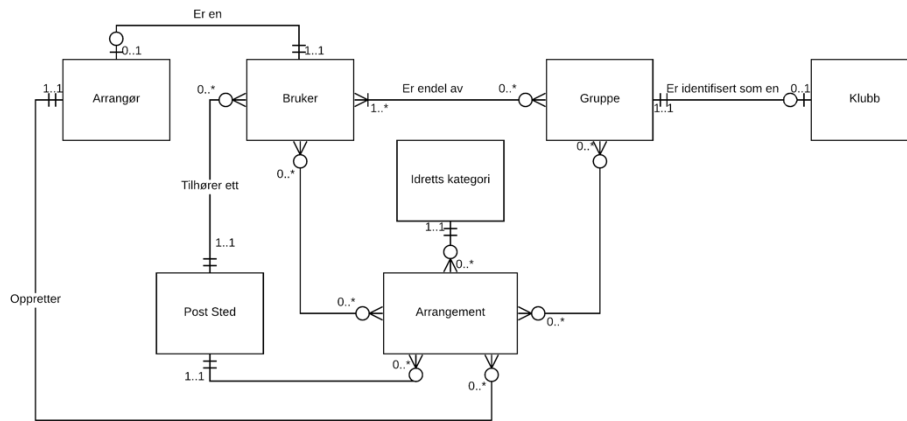
Database – Lagring av data

Lagring av atomær informasjon relatert til de ulike entitetene i systemet utføres i denne modulen. Videre vil denne modulen stå for mulighetene til å finjustere databasen, slik at den opererer på en mest mulig effektiv måte i henhold til kravspesifikasjonen.

Sensitiv data som brukerinformasjon blir lagret på en egen server for å høyne sikkerheten i systemet. Separert fra personinformasjonen vil mindre konfidensiell data lagres.

Struktur

Denne strukturen viser overordnet kjente relasjoner mellom de ulike entitetene i systemet. Videre vil denne informasjonen danne grunnlaget for en minimal konseptuell modell av databasen. Merk at dette ikke er en endelig modell, men en modell åpen for tolkning.



Figur 3 - Dette viser abstrakt, handlingsmønsteret ment for systemet

Prototype

Formålet med prototypen

Prototypen tar for seg oppgaven med å fremvise det mest prominente med systemet. Dette gjelder de følgende hovedfunksjonalitetene av systemet. Valgte funksjoner er ment for å definere og skape et mer helhetlig bilde av hva systemet tilbyr.

Valgt teknologi

Språket som prototypen er basert på er Java, der plattformen er [JavaFX](#) for selve brukergrensesnitt-funksjonene. Dette ble valgt da Java er kjørbart på de fleste datamaskiner som finnes, noe som medfører at det kreves få til ingen ekstra steg i utviklingsprosessen for prototypen. Videre ble det benyttet en integrasjon med [JUnit](#) biblioteket for testing av prototypen.

For å teste brukergrensesnittfunksjonalitet blir rammeverket *TestFX* brukt. En robot vil forsikre praktisk funksjonalitet i grensesnittet ved å simulere en *bruker* (knappetrykk, fylle inn inputfelter etc.) i systemet.

I tillegg ble det benyttet Maven for en enkel og kjapp integrering opp mot nødvendige biblioteker. Her sørget Maven for å både compilere og eksekvere applikasjonen, samt for selve bygg-funksjonaliteten av applikasjonen.

Lagring av data blir gjort i JSON-filer. Grunnen til det er for å unngå komplikasjoner ved å kjøre en database lokalt. Dette medfører at prototypen ikke er skalerbar ved lagring og uthenting av data.

Disse integrasjonene og teknologiene medførte at utviklingstiden ble kortere.

Valgte brukertyper

Brukertypene er valgt ut i henhold til hva som er mest relevant i det ferdige systemet.

[Bruker](#) er den brukertypen som faktiske *brukere* (deltagere) av arrangementer i systemet vil benytte seg av. Denne er med for å kunne vise frem funksjonaliteter vanlige personer har i systemet.

[Arrangør](#) er valgt ut for å få frem hvordan et idrettsarrangement vil bli opprettet og videre behandlet. Denne brukertypen demonstrerer den mest grunnleggende funksjonaliteten i systemet.

[Admin](#) er valgt ut for å demonstrere hvilke rettigheter en *serviceadministrator* vil ha i den ferdige versjonen av systemet.

[Gruppeleder](#) er valgt ut for å vise frem funksjonalitet som omhandler oppretting og administrering av grupper.

Valgte funksjonelle krav

Prototypen har som formål å representere de mest prominente kravene i henhold til kravspesifikasjonen. Krav i prototypen hvor det oppstår avvikelser fra de originale funksjonelle kravene vil refereres til *Svakheter i prototypen*.

Funksjonelle krav demonstrert i prototypen:

Link til segment Disse trenger linker/bokmerker	Valgte krav (Gitt med punkt)
Bruker.Brukergrensesnitt.Skal kunne se	1, 2, 3, 4, 6, 7
Gruppeleder.Brukergrensesnitt.Skal kunne se	1
Arrangør.Brukergrensesnitt.Skal kunne se	3
Serviceadministrator.Brukergresnesnitt.Skal kunne se	2
Alle brukere.Innlogging.Skal kunne logge inn	1 – Svakheter i prototypen: 1
Alle brukere.Sorterer arrangementer.Skal kunne	5, 6
Alle brukere.Søker arrangement.Skal kunne søke arrangementer	1, 2
Bruker.Påmelde arrangement.Skal kunne	1
Bruker.Påmelde arrangement.Skal ikke kunne melde på arrangementer	1, 3, 4
Bruker.Avmelde arrangement.Skal kunne melde av arrangementer	2
Bruker.Avmelde arrangement.Skal ikke kunne melde av arrangementer	3
Bruker.Opprette gruppe.Skal kunne	1 – Svakheter i prototypen: 4
Klubbleder.Administrerer klubb.Skal kunne	1, 2, 3, 4
Arrangør.Opprette arrangement.Skal kunne	1
Arrangør.Opprette arrangement.Må velge	1, 2, 3, 4, 5, 6, 12
Arrangør.Administrere arrangement.Skal kunne	1, 2
Arrangør.Administrere arrangement.Skal ikke kunne	3, 4
Serviceadministrator.Skal kunne	2, 3
Serviceadministrator.Skal ikke kunne	1, 2
System.Innlogging.Skal	1
System.Arrangør.Skal	3
System.Bruker.Skal	6
System.Gruppeleder.Skal	1

Valgte ikke funksjonelle krav

Link til segment	Valgte krav (gitt med punkt)
------------------	------------------------------

Ytelses relaterte krav	1, 2
------------------------	------

Data Definisjon

Beskrivelse

Dette er definisjonen av de ulike dataene som knyttes opp mot objekter. Derav også hva disse inneholder samt kjente restriksjoner.

Bruker

Data variabel	Format	Restriksjoner
Fornavn	Sekvens med Unicode bokstaver	Unicode karakterer som omfatter normal bruk av bokstaver. Maks lengde 255.
Etternavn	Sekvens med Unicode bokstaver	Unicode karakterer som omfatter normal bruk av bokstaver. Maks lengde 255.
Fødselsdato	YYYY-MM-DD	Maks alder tilsvarer 120år pr dags dato
Sted	Int-int-int-int	Firesifret tall
Adresse	Bokstaver-tall	Maks 100 bokstaver
e-post	navn@tjeneste.domene	Maks antall karakterer: 160
passord		Maks antall karakterer: 60
Bruker ID	ID betegnelse	ID generert av DBMS

Innlogging

Datavariabel	Format	Restriksjoner
e-post	navn@tjeneste.domene	Maks antall karakterer: 160
passord		Maks antall karakterer: 60

Arrangement

Data variabel	Format	Restriksjoner
ID	Unik ID	Generert av UUID
Navn	Karakterer i Unicode	30 > Antall karakterer > 2
Antall plasser	Integer	Tall < 1 000 000
Stednr	Int-int-int-int	Firesifret tall
Adresse	Bokstaver-tall	Maks 100 bokstaver
Start Dato	YYYY-MM-DD	
Slutt Dato	YYYY-MM-DD	
Start klokke	hh:mm:ss	ss settes lik 0 ved hver input
Slutt klokke	hh:mm:ss	ss settes lik 0 ved hver input
Gruppe	Booleansk verdi.	
Beskrivelse	Karakterer i unicode	Maks antall karakterer er 500

Idretts kategori	Navn - Egengenerert ID	Pre definert liste fra database
------------------	------------------------	---------------------------------

Grupper

Data variabel	Format	Restriksjoner
Navn	Sekvens med Unicode bokstaver	30 > Antall karakterer > 2
ID	Unik ID	ID generert av UUID
Medlemmer	En liste med objekter	

Idrettskategorier

Kategori	ID
Fotball	F-01
Basketball	B-01
Annet	A-01
Friidrett	F-01
Sykkelritt	S-01

Styrker

Følgende er kjente styrker, gitt på listeform:

1. Hele core er løst koblet opp mot resten av applikasjonen. Videre kan deler av core enkelt byttes ut uten at dette skaper noen store endringer, ettersom store deler av core baserer seg på interfaces.
2. GUI komponenter avhenger av core sine elementer, og av MainJavaFX som er å ansees som hovedcontrolleren.
3. Factory pattern er benyttet for DataFactory, da denne generer data, gitt en klasse.
4. Repository benytter reflection for å finne alle subclasses av IBasentity, som er basis interfacet for å beskrive data entitet, og etterspør HandleStorage for å hente ut data relatert til den entiteten, i form av en liste. Denne listen blir da mappet mot selve klassen som lagres statisk i Repository. Dette tillater å uthente informasjon basert på klasse og innlogget bruker.
5. Strukturen på lavnivå metoder har god struktur og er lesbar med oversiktlige metodenavn. Hver metode har kun en oppgave som skal utføres.
6. Alle klasser har en overordnet oversikt i form av en "guide" slik at det skal være lett å finne ut hvilke deler som utfører hva i koden.

Svakheter

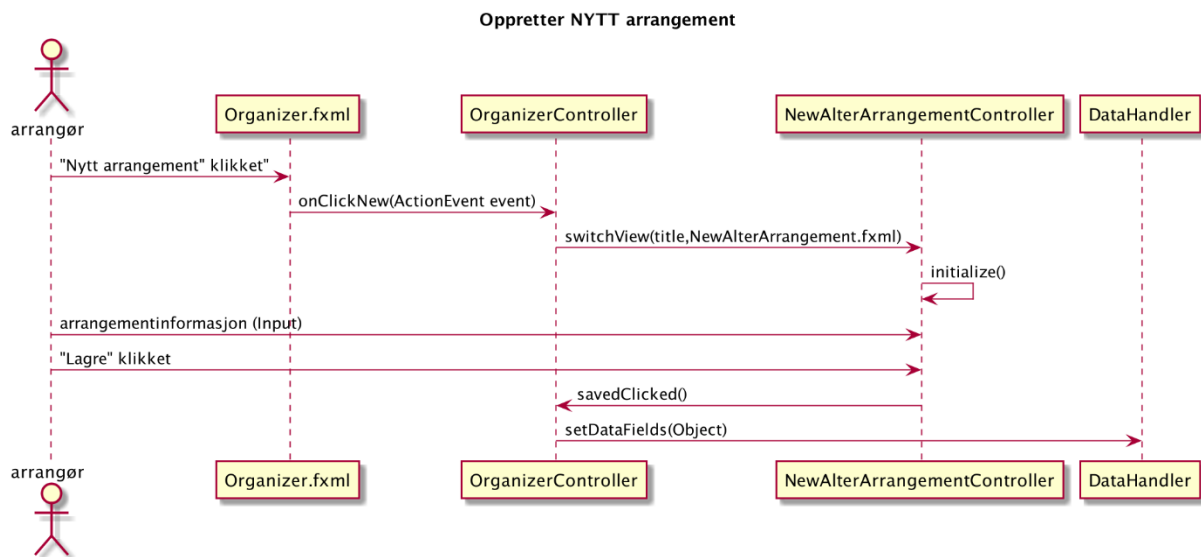
Følgende er kjente svakheter, gitt på listeform:

1. Prototypen kan kun håndtere en bruker fra hver brukertype. Dette vil si at lagring av data til fil må utbedres med en database som kan håndtere koblingen mellom mange brukere og mange arrangementer.
2. Grensesnittet er laget med biblioteket JavaFX. Dette biblioteket egner seg ikke til bruk for systemet da dette biblioteket ikke er webbasert.
3. Arrangementer inneholder ikke all informasjon da dette ikke har en endelig beskrivelse.

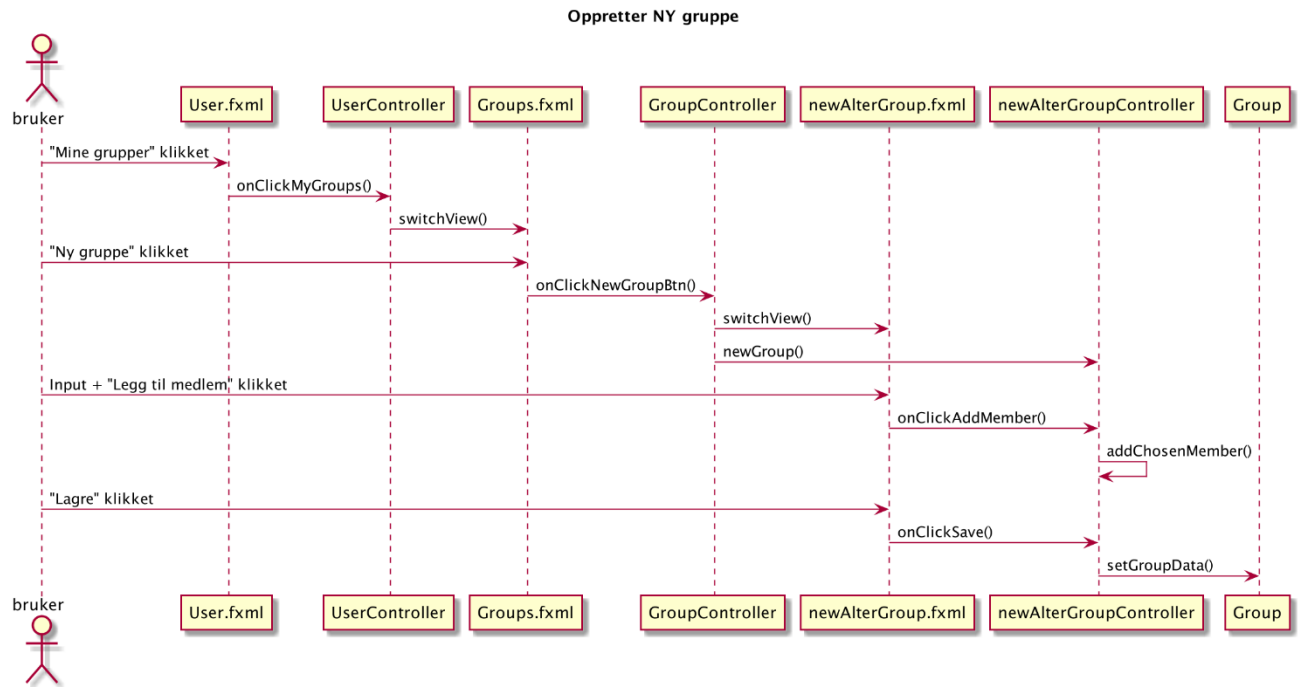
4. Gruppemedlemmer er ikke faktiske brukere, disse er bestående av endelige konstanter. Følgelig fører dette til redusert funksjonalitet.
5. Grupper kan ikke meldes på arrangementer. Funksjonen er utelatt for å minke arbeid med prototypen. I tillegg er denne funksjonen fremvist gjennom individuelle brukere.
6. Controllere har stor mangel på abstraksjon, noe som videre fører til mye redundans oppførsel og gjentakende funksjoner på kryss av disse. De har et mindre fokus da dette senere skal bli byttet ut med et webgrensesnitt i produksjon.
7. Ved opprettelse av et arrangement vil ikke listen hvor det velges "individuell konkurranse" og "lagkonkurranse" ha noen verdi. Alternativene er lagt til for å senere enkelt kunne ekspandere med denne funksjonaliteten.

Sekvensdiagram – Prototype

Viser flyten for *arrangør* som oppretter et arrangement:



Viser flyten for *bruker* ved oppretting av ny gruppe:



Tester

Disse testene er ment for å teste at systemet er bygd i henhold til valgte funksjonelle krav. Følgende tester er definert i prototypen.

UseCase

Disse tester at dataintegritet blir opprettholdt og at brukerfunksjoner utføres og gir forventet resultat. Under er det gitt informasjon om hvilke funksjoner som testes og hva det forventede svaret er:

Opprettelse av arrangement:

tests.notGUI.useCase.OrganizerUseCases# first_add_arrangement_and_keepDataIntegrity

- Genererer nytt arrangement som sendes videre til Repository for lagring. Videre testes det at det er mulig å hente ut samme arrangement gitt en ID som da sjekkes for at forventet informasjon har blitt lagret.

Slette arrangement:

tests.notGUI.useCase.AdminUseCase#delete_arrangement_keepDataIntegrity

tests.notGui.useCase.OrganizerUseCases#then_delete_arrangement_and_keepDataIntegrity

- Tester at et gitt arrangement blir slettet, samt at ingenting ekstra vil bli slettet. Videre tester dette at koblingene mellom påmeldte *brukere* og arrangementet vil bli slettet.

Rediger arrangement:

tests.notGUI.useCase.AdminUseCases# mutate_arrangement_and_keepDataIntegrity

- Tester at mutering av data tilhørende ett arrangement utføres. Dette innebærer også å teste for at uthentet arrangement faktisk har fått lagret sine endringer gjennom å hente det tilbake og sjekke det på nytt.

Påmelding av bruker for arrangement:

tests.notGUI.useCase.ProtoUserUseCases#first_insert_userRelation_to_arrangement_and_keepDataIntegrity

- Kobling mellom en *bruker* og et arrangement opprettes ved å registrere *brukeren* til arrangementet. Videre testes det at bare denne koblingen blir opprettet, og ingen nye koblinger har uforventet blitt opprettet.

Slette arrangement:

Tests.notGUI.useCase.AdminUseCase#delete_arrangement_keepDataIntegrity

Tests.notGui.useCase.OrganizerUseCases#then_delete_arrangement_and_keepDataIntegrity

- Tester at et gitt arrangement blir slettet, samt at ingenting ekstra vil bli slettet. Videre tester dette at koblingene mellom påmeldte *brukere* og arrangementet vil bli slettet.

Opprettelse av gruppe:

Tests.notGUI.useCase.ProtoUserUseCases#first_insert_new_group_and_keepDataIntegrity

- Genererer en ny gruppe som sendes videre til Repository for lagring. Her utføres det integritets tester for å sjekke at ingen ny informasjon har blitt lagret og ingen informasjon har blitt slettet.

Rediger gruppe:

Tests.notGUI.useCase.ProtoUserUseCases# then_mutate_group_with_newMembers_and_keepDataIntegrity

Tests.notGUI.useCase.ProtoUserUseCases#then_mutate_group_with_removedMembers_and_keepDataIntegrity

- Tester at mutering av data tilhørende en gruppe utføres.

Slette gruppe:

Tests.notGUI.useCase.ProtoUserUseCases# andLast_delete_group_and_keepDataIntegrity

- Tester at en gitt gruppe blir slettet, samt at ingenting ekstra vil bli slettet.

Unit Testing

Disse tester primært enkeltstående funksjonalitet, som eksisterer i ulike klasser. Disse er gruppert under funksjonalitet.

Tilgangs validering:

Tests.notGUI.unitTesting.AccessValidation# assert_Repository_UserLogin_Failed

- Tester at bare arrangører kan opprette arrangementer.
- Tester at bare de som er en arrangør og har opprettet det arrangementet, kan slette nevnt arrangement.

Innlogging av bruker:

tests.notGUI.thirdPartyIntegration.TestPaymentIntegration#assert_Repository_UserLogin_IsSuccessful

- Gitt informasjon korresponderer til predefinert *bruker* av systemet, samt informasjonen som sendes tilbake korresponderer med det forventede.
- Ved ugyldig informasjon, kastes det en exception og ingen annen informasjon sendes ut fra systemet.

System funksjonalitet - validering:

tests.notGUI.unitTesting.SystemFunctionality#assert_arrangement_has_requiredData

tests.notGUI.unitTesting.SystemFunctionality#assert_DateTest_validates_legal_dates

tests.notGUI.unitTesting.SystemFunctionality#assert_DateTest_validates_illegal_dates

tests.notGUI.unitTesting.SystemFunctionality#assert_DateTest_validates_sorts_correct

tests.notGUI.unitTesting.SystemFunctionality#assert_Validation_ofNumber_isIllegal

tests.notGUI.unitTesting.SystemFunctionality#assert_Validation_ofNumber_isLegal

- Validerer at arrangement klassen inneholder minimal informasjon, som benyttes av prototypen.
- Verifiserer at klassen DateTest utfører validering av lovlige og ulovlig verdier for en dato.
- Verifiserer at klassen DateTest utfører korrekt sortering av statiske lister.
- Verifiserer at klassen Validation utfører korrekt validering av ulovlige og lovlig verdier gitt input av en klasse.
- Verifiserer at klassen Validation utfører korrekt validering av lovlige og ulovlige verdier for tall.

Arrangement:

tests.notGUI.unitTesting.ArrangementValidationOf.assert_Arrangement_Input_isLegal

tests.notGUI.unitTesting.ArrangementValidationOf.assert_Arrangement_InputNamesAndDates_isIllegal

Det testes for at opprettelse av gruppe er valid ved lovlig navn, og invalid gitt ulovlige verdier.

- Opprettelse gitt lovlig navn er mulig
- Ikke mulig å opprette gitt ulovlige navn

Grupper:

tests.notGUI.unitTesting.GroupValidationOf#assert_factory_Input_isLegal

tests.notGUI.unitTesting.GroupValidationOf#assert_Group_Name_isIllegal

Det testes for at opprettelse av gruppe er valid ved lovlig navn, og invalid gitt ulovlige verdier.

Funksjoner som testes er:

- Opprettelse av gruppe gitt lovlig navn er mulig
- Ikke mulig å opprette gitt ulovlige gruppe navn

Bruker grensesnitt

Dette er tester som omhandler brukergrensesnittet og dets funksjonalitet. Her er det brukt rammeverket TestFX. Et rammeverk som tillater å simulere interaksjon mellom GUI og bruker av applikasjonen.

Det ble valgt å innføre minimale GUI tester, da denne delen av applikasjonen ofte var under refaktorering eller utvidelse. Dette førte til at det ble både tungvint og tidkrevende å utføre manuell testing av funksjonaliteten.

NB: Når GUI-testene kjører vil testene ta kontroll over tastatur og mus på datamaskinen. Her er det ekstremt viktig å ikke berøre nevnte enheter. Dette vil kunne medføre at testene ikke opptrer på en optimal måte. Mulig bivirkning av interaksjon under kjøring, kan føre til at tester feiler, uten at det nødvendigvis er feil i testene.

Boksene i diagrammene representerer klikkbare elementer i brukergrensesnittet.

Hver test metode er valgt å strukturere som en sekvens med tester, som representerer interaksjon på GUI-et. Etter hvert steg, vil det testes for at gitt input sekvens fører til en ønsket tilstand. Dette kan være ting som: at riktig klikk fører til riktig vindu, riktig felter fylles ut med forventet informasjon gitt en interaksjon. Under er det gitt use case scenarioer hentet fra brukergrupper. Strukturert slik at hvert steg er ett trinn, i en sekvens med handlinger på GUI-et:

Serviceadministrator:

tests.GUI.controller.AdminController# select_Arrangement_AndEdit

- Velger et arrangement
- Klikker rediger
- Endrer informasjonen
- Lagrer utførte endringer
- Sletter arrangement

Arrangør

tests.GUI.controller.OrganizerController#assert_new_arrangement_is_created

- Klikker nytt arrangement
- Skriver inn gyldig informasjon
- Lagrer dette
- Klikker på opprettet arrangementet
- Klikker rediger
- Redigerer med gyldig informasjon
- Lagrer endring
- Slett arrangementet.

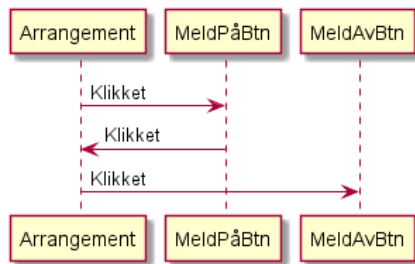


Bruker

tests.GUI.controller.OrganizerController#add_and_remove_user_on_arrangement

- Klikker på ett tilgjengelig arrangement
- Klikker på meld på

- Klikker på påmeldt arrangement
- Klikker meld av – På registrerte arrangementet.



Gruppeleder

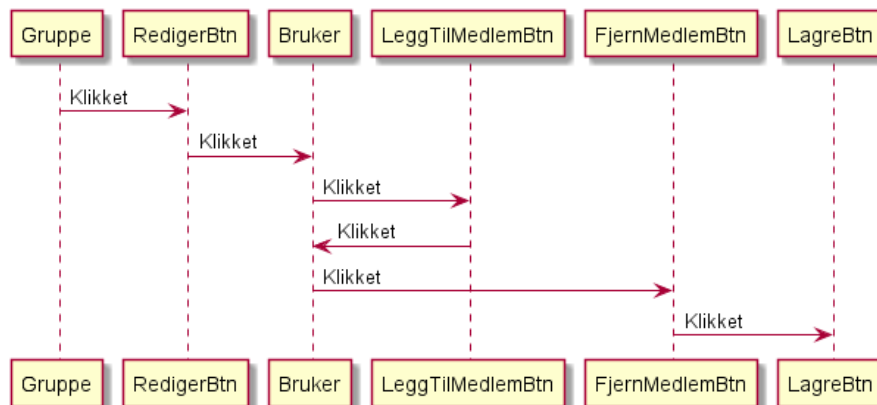
tests.GUI.controller.GroupController# assert_Group_isSelectedAndEdited

- Klikker på en opprettet gruppe
- Klikker rediger
- Klikker på tilgjengelig bruker
- Klikker Legg til
- Klikker på lagt til bruker
- Klikker på fjern bruker
- Klikk lagre

Login

tests.GUI.controller.MainController#enter_correct_login_information_from_btns

For hver brukertype vil gitte Bruker type logges inn og deretter ut. Dette utføres med alle de tre pre-definerte *brukerne*. Poenget med denne testen er for å sikre at riktig bruker informasjon åpner korresponderende vindu.



Betalingsintegrasjon stubb

Klarna

tests.notGUI.thirdPartyIntegration.TestPaymentIntegration#klarna_access_success

tests.notGUI.thirdPartyIntegration.TestPaymentIntegration#klarna_payment_accepted

Tester om koblingen til Klarna gir forventet respons: 200(Godkjent HTML-forespørsel)

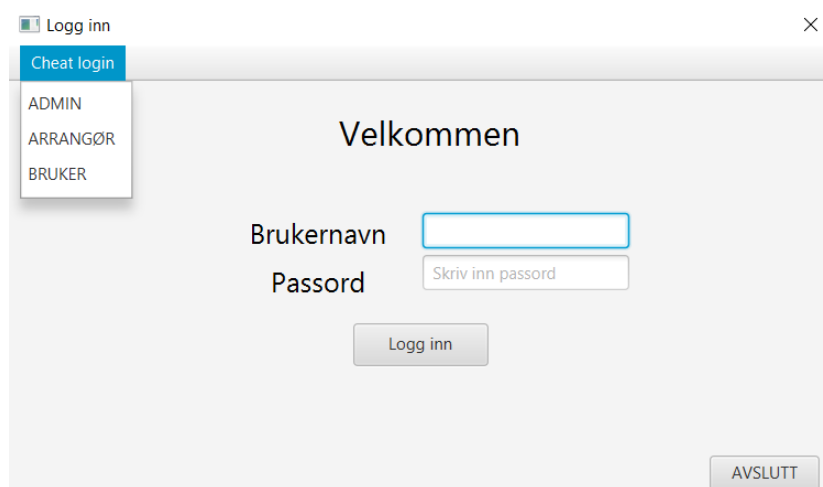
Tester validering av betaling ved å anta at man får ett resultat fra Klarna for deretter håndtere dette og gi et sant/usant svar tilbake til systemet.

Forventet respons fra Klarna: Godkjent = "Accepted", Avvist = "Declined"

Brukermanual for Prototype

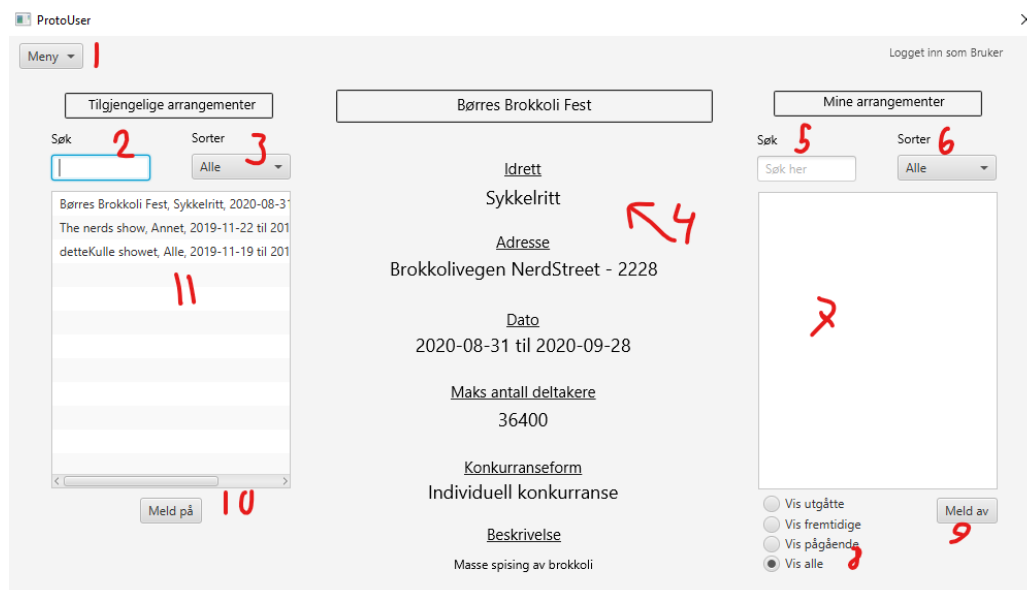
Login

Første vindu du møter er login-vinduet. Her skriver du inn brukernavn og passord og deretter klikker «logg inn». For prototypen er det tre allerede definerte *brukere* med brukernavn og passord. Disse finner du øverst til venstre ved å klikke på menyvalget «Cheat login».



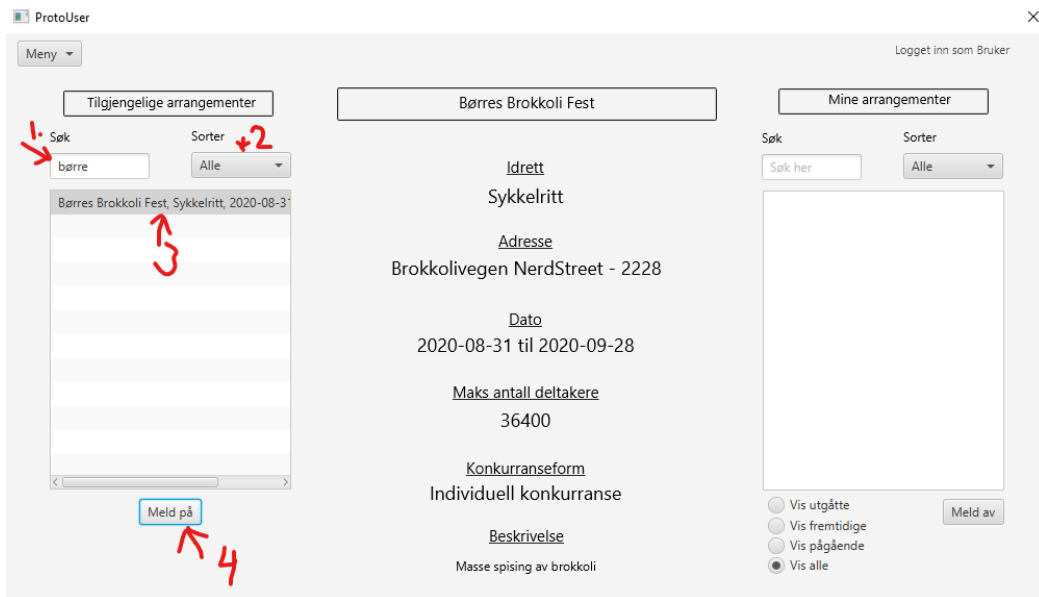
Login som bruker

Første vindu etter å ha logget inn som *bruker* er en oversikt over alle tilgjengelige arrangementer på venstre side (11), alle påmeldte arrangementer på høyre side (7) og informasjon om valgt arrangement i midten (4). Du har mulighet til å søke og sortere i arrangement-listene ved å benytte deg av søkefeltene, sorteringsknappene og radioknappene 2, 3, 5, 6 og 8.



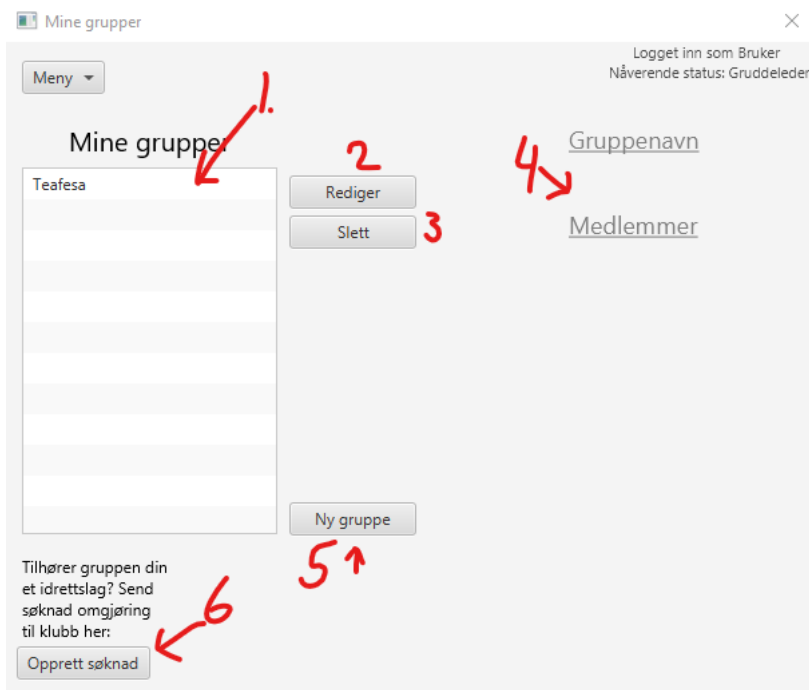
Påmelding

For å melde deg på et arrangement finner du først ønsket arrangement via søke- og sorteringsfunksjonene (1, 2), velger ønsket arrangement (3) og deretter trykker «meld på» (4).



Grupper

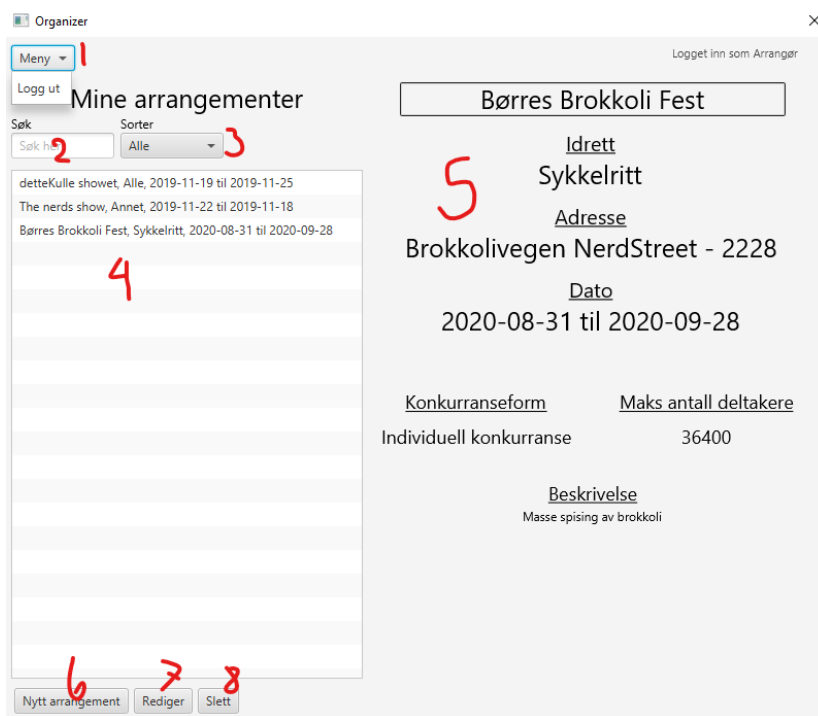
Som en *bruker* kan du opprette egne grupper. Ved å velge «Meny» øverst i venstre hjørne, og deretter «Mine grupper», får du opp gruppevinduet.



På venstre side har du en liste over dine grupper (1), og informasjon om valgt gruppe på høyre side (4). Her har du mulighet til å slette og redigere gruppene dine (2, 3), eller opprette en ny gruppe (5). Om du tilhører en idrettsklubb og ønsker å sende inn en søknad, benytter du deg av «Opprett søknad» (6).

Login som arrangør

Første vindu etter å ha logget inn som arrangør, er en oversikt over arrangementene dine, med en liste over alle arrangementer på venstre side (4) og informasjon om valgt arrangement på høyre side (5).



Som arrangør har du muligheten til å slette (8), redigere (7) eller opprette et nytt arrangement (6), og sortere arrangementlisten (4) ved å benytte deg av søkefeltet og sorteringsknappene (2, 3).

Login som admin

Første vindu etter å ha logget inn som admin er en oversikt over alle arrangementer til alle arrangører i en liste på venstre side (3) og informasjon om valgt arrangement på høyre side (4). Du har mulighet til å søke etter spesifikke arrangører med søkefeltet (2), eller redigere og slette arrangementer (5, 6).

Admin

Meny Status: Online

Logget inn som serviceadministrator

Søk her

Børres Brokkoli Fest, Sykkellritt, 2020-04

The nerds show, Annet, 2019-11-17 til

detteKulle showet, Alle, 2019-11-19 til

Idrett

Adresse

Dato

Konkurranseform

Maks antall deltakere

Beskrivelse

Rediger Slett

Hvordan kjøre prototypen

Denne prosessen er gitt stegvis:

- Enten høyreklikk i mappen og velg open with intellij, eller åpne IDE og velg open project, etterfulgt av å velge prosjektmappen.
- Velg importer maven og vent for denne til å sette opp og laste ned dependencies.
- Bruk en av de predefinerte konfigurasjonene for å utføre en handling
- Main kan kjøres enten gjennom predefinerte funksjoner eller ved å Høyreklikke på klassen "Main", deretter klikk "Run".

Ordforklaringer

JAVAFX – Dette er et bibliotek som blir brukt til utvikling av programvare. Biblioteket brukes til å utvikle vindusbaserte grensesnitt.

TESTFX – Dette er et bibliotek som blir brukt til automatisert testing av brukergrensesnitt som er basert på JavaFX.

MAVEN – Verktøy for å samle og håndtere eksterne avhengigheter som måtte fremkomme i et prosjekt.

JSON – Et filformat ment for lagring av data på en strukturell måte.

SERVER – Dedikert maskin som står for utførelse av tjenester som videre delegeres ut mot *brukere* av systemet. Eg: En webserver som serverer nettsider til en *bruker*.

SERVER SIDE – programvare lokalisert på en server som utfører tjenester for systemet.

DATA – Generell betegnelse for informasjon om objekter.

DATABASE – Samlebetegnelse for lagringsmedium av større mengder med data.

DBMS – Database Management System – System for håndtering og sikring av data, primært databasen.

XSS – (Cross Site Scripting) – Resultatet av at en *bruker* skriver inn skadelig data i tekstbokser.

JUNIT – Et bibliotek som blir brukt i programvare utvikling primært for å validere, samt verifisere systemets funksjoner.

INT – Ett naturlig heltall. Eks: {...-5, -4, -3, -2, -1, 0, 1, 2, 3, 5.....}

STRING – Samling av tegn som til sammen utgjør en tekst. Eks: "Hei" eller ".0-9a-zæøå ' ^".

Boolean – En verdi som sier om noe er sant eller usant. Representert binært som 0 eller 1.

NULL – Tom verdi. I binært er dette en verdi representert der alle tegnene er null.

LocalDate – *Datoformat(dd.mm.åå)*

Exception – Unntakssituasjon i koden/filen til et program.